395

Data-Writer™

by David S. Walonick
An Information Processing System

User's Manual

TRS-80 Model I/III (48K)

Published by

SOFTWARE OPTIONS, INC.
PO Box 970
Bowling Green Station
New York, NY 10274

SOFTWARE OPTIONS INC ™

Copyright and Trademark Notice

DATA-WRITER user's manual and programs copyright © 1982 David S. Walonick. All rights reserved. Any duplication of these programs in whole or in part, except for the making of needed backup copies for sole personal use, without express written permission from David S. Walonick c/o Software Options is a copyright violation and is absolutely prohibited.

The runtime library for compiled programs is copyrighted by Microsoft, Inc. TDOS is copyrighted by Micro-Systems Software, Inc.

DATA-WRITER is a trademark of Software Options, Inc., P.O. Box 970, Bowling Green Station, New York, N.Y. 10274

How to Install

Data-Writer™

Hardware Requirements

DATA-WRITER requires a TRS-80 Model I or Model III with 48K user memory and at least two disk drives. A Model I must have lower case. DATA-WRITER supports a hard disk.

The instructions provided on the following pages will allow you to install DATA-WRITER on your system with little effort.

Regardless of the version of DATA-WRITER you have received, the first thing you should do after reading the appropriate installation instructions is to make a backup copy of each program diskette. DATA-WRITER program diskettes are not copy-protected, so you will be able to make as many backup copies as you need for your own use. Making copies for others is strictly prohibited by U.S. copyright laws. (If you like DATA-WRITER and you think your friends would find it useful, please encourage them to purchase their own copies. We will be able to continue to provide support for and enhancement to our programs.)

A Note for Model III TRSDOS Users

Because of incompatibilities with Microsoft's compiler, Model III TRSDOS users have been supplied with a DOSPLUS version of DATA-WRITER and instructions for converting programs and data files to DOSPLUS format. See pages ix-xii.

Model I Users

You have received the Model I version of DATA-WRITER on three 35-track single-density diskettes in TRSDOS format. These diskettes do not contain an operating system, but you can copy DATA-WRITER to system diskettes using the instructions given below. This version of DATA-WRITER will work with TRSDOS (single-density only), DOSPLUS, LDOS (single-density only), NEWDOS 2.1, NEWDOS/80 V2.0 and DBLDOS.

There are ten programs in the DATA-WRITER system. All programs have been compiled with the Microsoft BASIC compiler, and therefore all program names end with "/CHN" for "chain." For example, the DATA-WRITER form letter program is called LETTERS/CHN.

Each DATA-WRITER program has a special "loader" program that takes care of certain housekeeping tasks. All loader program names end with "/CMD." For example, the loader for the form letter program is called LETTERS/CMD.

Both the loader program and the chain program must be on your diskette to run any DATA-WRITER program. Also the BRUN/CMD file must be on-line. This is a runtime library for compiled programs and must be on each DATA-WRITER program diskette.

Therefore, to run any DATA-WRITER program you must have these three programs on your diskette:

PROGRAM/CHN
PROGRAM/CMD
BRUN/CMD

where "PROGRAM" is the name of a DATA-WRITER program.

You'll probably want to copy the DATA-WRITER programs onto DOS diskettes. Then the DATA-WRITER programs can always reside in Drive 0, and your data files can be on Drive 1 or any other drive. (DATA-WRITER will work regardless of which drive contains the programs.)

How to Make a Minimum System Diskette

The DATA-WRITER programs are too large to fit on one system diskette. In fact, you will need three or four diskettes to hold all the DATA-WRITER programs and your DOS. If you will operate in single density, then you should make four copies of your minimum operating system. If you will use double density then make three copies of your minimum operating system. A minimum operating system is one from which all non-essential files have been eliminated.

To make a minimum operating system, make a backup of your operating system diskette. Then kill all files that are not essential to DATA-WRITER's operation (as described below). The following list tells what files you should leave on your disk. Check your operating system manual for a full list of system files. Then kill all files except the ones listed below. *Note:* TRSDOS users should kill all the files listed on the TRSDOS directory, plus BASIC/CMD and BASICR/CMD. Use the commands "KILL BASIC/CMD. BASIC" and "KILL BASICR/CMD. BASIC".

TRSDOS, NEWDOS 2.1 and DBLDOS
- **BOOT/SYS**
- **DIR/SYS**
- **SYS0/SYS through SYS6/SYS**

DOSPLUS
- **BOOT/SYS**
- **DIR/SYS**
- **SYS0/SYS through SYS8/SYS**

NEWDOS/80 V2.0
- **BOOT/SYS**
- **DIR/SYS**
- **SYS0/SYS through SYS4/SYS**
- **SYS6/SYS through SYS8/SYS**
- **SYS14/SYS**
- **SYS16/SYS**
- **SYS17/SYS**

LDOS (Single density only)

BOOT/SYS
DIR/SYS
SYS0/SYS through SYS4/SYS
SYS7/SYS through SYS8/SYS
SYS10/SYS

After creating a minimum operating system, you should have at least 48 free grans on your diskette. If you do not, you have not killed all non-essential files.

How to Copy DATA-WRITER to your System Diskettes

After you have made several backup copies of your minimum operating system, you can transfer the DATA-WRITER programs to these diskettes, following the instructions provided below. You will need to decide which programs you want to put on each diskette. You can copy any DATA-WRITER program to any of your minimum system diskettes with only two constraints:

1) BRUN/CMD must be on each diskette.
2) Both the PROGRAM/CMD and the PROGRAM/CHN file must be the same diskette. ("PROGRAM" is any DATA-WRITER program.)

All the /CMD loader programs are one gran in length. BRUN is 12 grans long. The /CHN programs are the length indicated:

ENTRY/CHN	**14 grans**
EDIT/CHN	**12 grans**
MANAGE/CHN	**15 grans**
STATS/CHN	**6 grans**
MATH/CHN	**13 grans**
SELECT/CHN	**8 grans**
SORT/CHN	**12 grans**
LABELS/CHN	**12 grans**
LETTERS/CHN	**16 grans**
REPORT/CHN	**19 grans**

One way to set up a single-density system would be:

DISK 1	DISK 2
BRUN/CMD	BRUN/CMD
ENTRY/CHN	MANAGE/CHN
ENTRY/CMD	MANAGE/CMD
EDIT/CHN	REPORT/CHN
EDIT/CMD	REPORT/CMD
STATS/CHN	
STATS/CMD	

DISK 3	DISK 4
BRUN/CMD	BRUN/CMD
SORT/CHN	LETTERS/CHN
SORT/CMD	LETTERS/CMD
SELECT/CHN	LABELS/CHN
SELECT/CMD	LABELS/CMD
MATH/CHN	
MATH/CMD	

If you are using TRSDOS or LDOS as your operating system, then you've now finished the installation, and you are ready to begin running the system. If you are using any other operating system, you need to patch BRUN/CMD so it will work with your system. BRUN/CMD must be patched on each of the DATA-WRITER minimum system diskettes you have created.

How to Patch BRUN

Insert a DATA-WRITER minimum system diskette in Drive 0 and the diskette containing our patch program (the first diskette) in Drive 1. From DOS, type

PATCH <ENTER>

The patch program will run and ask which drive contains the BRUN/CMD module to be patched. Answer "0", and the BRUN CMD module on the diskette in Drive 0 will be patched. Perform this patch on each DATA-WRITER system diskette you created. We recommend that you write-protect your program diskettes. You should also make backup copies of your patched system diskettes.

Model III Users

You have received one of three Model III versions of DATA-WRITER contained on two 40-track double-density diskettes.

DOSPLUS version

If you specified DOSPLUS compatibility when you placed your order, or if you purchased DATA-WRITER from a software dealer or did not otherwise specify operating system compatibility, then the DATA-WRITER diskettes you received contain a kernal of the DOSPLUS operating system called TDOS.

The entire DOSPLUS 3.4 operating system is much larger than the system we sent you. Micro-Systems Software, Inc. has given us permission to distribute DATA-WRITER with their DOSPLUS kernal. As you become more proficient with DATA-WRITER , you may want to purchase the entire DOSPLUS system.

TDOS will allow you to boot directly from either DATA-WRITER program diskette without first loading another operating system into your computer. You can format data diskettes, make backup copies, kill files, and perform other standard tasks without the use of a more complex operating system.

The concepts you learned in using TRSDOS are still applicable to DOSPLUS; however, there are some differences you should be aware of.

How to Boot from TDOS

Here's how to boot from the TDOS version of DATA-WRITER:

1. Turn on your computer
2. Place either DATA-WRITER diskette in Drive 0.
3. Press the orange RESET button of your computer.

Your drive will come on and its LED will glow. The DATA-WRITER and SOFTWARE OPTIONS logos will appear on your screen along with the list of DATA-WRITER programs on that diskette. The prompt,

DOS PLUS

will appear at the lower left of your screen. You may now type commands from your keyboard, as described below and in the *How to Use DATA-WRITER* section of this manual.

How to Format a DOSPLUS Diskette

When formatting a new data diskette for use with DATA-WRITER, you must use DOSPLUS. The DOSPLUS FORMAT procedure is nearly identical to the TRSDOS procedure except that a few extra questions will be asked. After booting, place the diskette to be formatted in Drive 1 and type

FORMAT

You will be asked the usual questions about drive number, format date and password. When DOSPLUS asks for the "number of cylinders," just press <ENTER> and it will default to 40. (If you have an 80-track drive, enter "80".) When DOSPLUS asks "single or double density," just press <ENTER> and it will default to double density.

How to Make Backups with DOSPLUS

The DOSPLUS BACKUP utility works like the TRSDOS version. The DATA-WRITER program diskettes are not copy-protected, so you will be able to make as many backup copies as you need for your own use. Making copies for others is strictly prohibited by U.S. copyright laws.

DIRectory and CATalog

The DIR (directory) command is not supplied by Micro-Systems Software with TDOS. Therefore, we have supplied our own DIR command that will display the names of the DATA-WRITER programs currently in Drive 0.

To see the data files that you have created that reside on Drive 1 or 2, type the DOSPLUS CAT (catalog) command as follows:

CAT :1 or CAT :2

DOSPLUS for TRSDOS Users

If you are a TRSDOS user who has programs and data files residing on TRSDOS formatted diskettes, and you want to use these programs, such as Scripsit, or data files with DATA-WRITER, just follow the instructions provided below to transfer them to DOSPLUS formatted diskettes.

DOSPLUS is not the same as TRSDOS, and the two systems are not compatible. You will not be able to mix DOSPLUS and TRSDOS diskettes, so we suggest you label your diskettes carefully.

Even though DOSPLUS and TRSDOS are incompatible, TDOS provides methods to call a TRSDOS directory and to transfer programs or data files from TRSDOS to DOSPLUS formatted diskettes. To look at the directory of a TRSDOS diskette contained in Drive 1 or 2 while using TDOS, use the following CAT command:

CAT :1 (T) or CAT :2 (T)

Transferring Programs from TRSDOS Diskettes

To transfer a file from a TRSDOS diskette to a DOSPLUS diskette, use the DOSPLUS CONVERT utility. You can CONVERT all the files on your diskette at once or just a single file.

To transfer all files on a TRSDOS version 1.3 diskette to a DOSPLUS diskette, type the following command:

CONVERT:s :d (V13)

where "s" is the number of the source drive and "d" is the number of the destination drive. To CONVERT all files on an earlier version of TRSDOS, type the same command but without the "(V13)". No version number is required for earlier TRSDOS versions. Be careful to observe the required spaces.

To transfer a single file from a TRSDOS version 1.3 diskette to a DOSPLUS diskette, use the following command:

CONVERT FILENAME:s :d (V13)

where FILENAME is the name of the file you want to transfer, "s" is the number of the source drive, and "d" is the number of the destination drive. To transfer a file from an earlier TRSDOS version diskette, just leave off the "(V13)".

Note: A diskette containing the TDOS system must be in Drive 0 during the conversion. If you have only two disk drives, you can use the following procedure to transfer one or all of your files to a DOSPLUS formatted diskette:

1. Make a backup of one of your DATA-WRITER program diskettes.

2. KILL all the DATA-WRITER programs on this backup diskette, using the KILL command. For example,

KILL ENTRY/CMD

Note: For every /CMD file on your DATA-WRITER backup diskette, there is also a /CHN file, which is invisible. These should also be killed to provide enough space to hold your files to be transferred.

3. This diskette will now have only the TDOS system on it, so we'll refer to it as the "TDOS system diskette." Put this diskette in Drive 0 and your TRSDOS diskette in Drive 1.

4. Type the appropriate CONVERT command as described above using Drive 1 as the source drive and Drive 0 as the destination drive. For example, to CONVERT Scripsit from a TRSDOS version 1.3 diskette to DOSPLUS format, type the following:

CONVERT SCRIPSIT/CMD:1 :0 (V13)

5. The converted file or files will now reside on your "TDOS system diskette" in Drive 0. To copy these files onto a formatted DOSPLUS data diskette, use the DOSPLUS COPY utility. Put a formatted DOSPLUS diskette into Drive 1 and type

COPY FILENAME:0 :1

The CONVERT utility will transfer files from TRSDOS diskettes to DOSPLUS diskettes but not the other way around.

This completes the installation of the DATA-WRITER system. Put the master diskettes we sent you in a safe place.

To use DATA-WRITER, most users insert the appropriate DATA-WRITER program diskette in Drive 0 and their data diskette in Drive 1. A data diskette is a formatted diskette that does not contain an operating system.

* * *

A Note on DATA-WRITER's ENTRY Program

If you have a recently manufactured Model III, you will not be able to delete a character by pressing just the <SHIFT> and <DOWN-ARROW> keys simultaneously, as explained on page 21. Radio Shack has defined these two keys as a control function in its revised Model III keyboard driver. Instead, to delete a character during data entry, press the <SHIFT> and <DOWN-ARROW> keys simultaneously and, while holding them down, press <Z>.

LDOS and NEWDOS/80 Versions

If you specified compatibility with LDOS or NEWDOS/80 when you placed your order, your DATA-WRITER diskettes are in LDOS or NEWDOS/80 format, but do not contain an operating system.

If you want to boot directly from an LDOS or NEWDOS/80 version DATA-WRITER diskette, follow the instructions provided below to copy DATA-WRITER onto system diskettes.

1. Make two backups of your LDOS or NEWDOS/80 master diskette. (These must be made from your original operating system diskette, not an abbreviated or pirated copy.)
2. Place one operating system backup in Drive 0 and the first DATA-WRITER program diskette in Drive 1. Reset your computer by pressing the orange RESET button.
3. If you are using NEWDOS/80, skip to step 5. If you are using LDOS, type

   PURGE :0 (I) <ENTER>

4. As LDOS asks whether you want to kill a file, answer "Y" for yes for every file. When you have finished you will have killed all non-essential files on your minimum system diskette..
5. For both NEWDOS/80 and LDOS systems, type

   DO TRANSFER <ENTER>

   The DATA-WRITER programs will automatically be copied to your minimum system diskette.
6. Remove both diskettes. Write-protect your system diskette and label it "DATA-WRITER".
7. Follow steps 2 through 6 again using your other operating system backup and the second DATA-WRITER program diskette.

This completes the installation of the DATA-WRITER system. Put the master diskettes we sent you in a safe place.

How to Use

Data-Writer™

Table of Contents

INTRODUCTION **5**

I. Of Storks and Ships and Data Files **9**

Data Base Organization **9**
Data in a Data Base **10**
Special Considerations for Field-Labels **12**
When to Use Your Word Processor **15**
Adding Records to Your Data Base **15**
Saving a Data File to Disk **16**
Reworking an Old File for DATA-WRITER Use **16**

II. ENTRY: Adding Data to Your Data Base **17**
III. EDIT: Making Changes to Your Data **22**
IV. MANAGE: Manipulating Your Data Base **28**
V. STATS: How to Verify Your Data Base **34**
VI. MATH: Using Math Procedures **37**
VII. SELECT: Making a Subset of Your Data Base **40**
VIII. SORT: Ordering Your Data Base **45**
IX. LABELS: Creating Mailing Labels **49**
X. LETTERS: How to Create a Form Letter **52**
XI. REPORT: How to Create a Report **67**

TECHNICAL SUPPORT **81**

LIMITED WARRANTY **83**

REGISTRATION FORM **85**

Introduction

DATA-WRITER is a data base manager that can be used with your word processor or by itself as a complete stand-alone system for managing textual and numeric data. There are ten powerful programs in the DATA-WRITER system that provide the user with a full set of functions for managing and using data bases, including the ability to change their structures. Mailing list maintenance, operational reporting, form letter production to a large list or a subset, and hundreds of other tasks are all easily done with DATA-WRITER.

Here is a brief description of the ten DATA-WRITER programs, each of whose functions is described in detail in later chapters of this manual:

Data Entry Programs

1. ENTRY lets you create a new data base or add new records to an existing data base without using your word processor. It has special features that enable you to perform validity checks on your data while the information is being entered. Special features include automatic repeating fields, abbreviation tables, numeric validity checks, mandatory entry checks, audible signal, and other special features to make data entry easier.

2. EDIT: Once your data base is created, you can use your word processor or DATA-WRITER's EDIT program to make changes. You can edit your data base even if it's too large to be edited by your word processor. EDIT has many features of a word processor, including full cursor controls and string search. Use EDIT in its "View Mode" to display a record for reference.

File Management Programs

3. Using MANAGE, you have the ability to restructure your data base without having to edit it manually. You can add new fields, delete fields, rearrange fields and append one field to another. MANAGE also gives you the ability to merge two data bases together or to split a very large data base into two smaller ones.

4. After you have entered or modified your data, you can use the STATS program to search your data base for inconsistencies. STATS prints or displays a statistical report showing you any format errors. You can show just the errors or list the entire data base. STATS tells you the number, location and nature of each error. It tells how many records there are in the file, the labels of your fields, and the maximum data length of each field.

5. MATH lets you define fields as equations. You'll be able to calculate any fields in your data base using addition, subtraction, multiplication, division and exponentiation. The formulas you create may freely mix field labels and numbers. You may use up to 20 equations per run, and each equation may contain up to 255 characters. In addition, there are ten temporary storage locations for storing intermediate results. This gives you the ability to perform very complex mathematical procedures. All math procedures are performed in double-precison arithmetic giving you accuracy to 16 places. You can store your equations on disk.

6. SELECT enables you to create subsets of your original file by specifying limiting criteria, such as SELECT IF SEX = F, or SELECT IF AMOUNT > 1000. Each subset is then stored under a new file name to be used as a new data base. This lets you write reports containing selected data, such as overdue accounts or low inventory-level items. Send form letters only to females in a data base, or just to the doctors. SELECT provides for tremendous flexibility in using your data base with the other programs in the DATA-WRITER system.

7. SORT is a fast, two-level sort program that lets you order a list of items into alphabetical or numerical sequence, in ascending or descending order. Sort on any field, without having previously designated it as a key. You can even sort by last name or zip code embedded in a line. SORT then creates a new file, with your records in the newly specified order.

Output Programs

8. LABELS: You can custom design a label format to print mailing labels from your data base. Print one to four labels across. Chain data bases to print several sets of labels. You may preset any line with a fixed message. Also, you may save a label format on disk, reuse it as often as necessary, and later modify it.

9. LETTERS: You can easily create a form letter into which you can insert, from your data base, up to 20 different names, addresses, amounts, keywords, even phrases. The date or other pertinent information can also be inserted from the keyboard. The LETTERS program has a text editor that lets you change an old form letter or create a new one, insert control codes into a previously-written letter, or make whatever changes you want. Your letter instructions can be stored on disk for use again and again.

10. REPORT is a program for writing reports, such as inventories, accounts payable and receivable, bibliographies, insurance coverage, stock issues...the list is endless. REPORT has a text editor that allows you to define or modify a report format. The use of page and columnar headings and control codes lets you format your data as you want. Then you can print your report according to your format. Your report format can then be stored on disk for reuse whenever you want.

* * *

This manual describes how to set up a data base and how to use DATA-WRITER's programs for many business and personal applications. DATA-WRITER is both powerful and easy to use. Reading this manual will help you get the most out of this remarkable system.

We recommend that you start with the first chapter, whether you intend to use your word processor to create your data base or not. Despite its cute title, the first chapter is important for understanding DATA-WRITER's data base organization requirements.

We have put considerable effort into making DATA-WRITER useful in solving many business and personal information processing problems. We welcome your suggestions, praise, or criticism about the programs and the manual, or information on how you're using them.

In any case, please return the *Registration Form* in the back of this manual so we can advise you of updates and new uses for DATA-WRITER.

Notes:

1. See the separately included installation instructions to use DATA-WRITER with your disk operating system (DOS) and word processor.
2. To load and run any DATA-WRITER program, you must mount the diskette containing the program you want to run, together with a diskette containing your data base, on your system.
3. In general, DATA-WRITER will support a data file, in proper format (see the first chapter), contained on a single diskette, regardless of how large it is. The SORT program will handle a data file of up to 4500 records.
4. In this manual we have used the convention < > to indicate a key that you should press on your keyboard. For example, <SHIFT> is the shift key.
5. Your data are undoubtedly important to you. We suggest that you create a diskette backup plan which provides for frequent backup of your data files.
6. If you haven't already done so, make a backup copy of your DATA-WRITER programs.

I. Of Storks and Ships and Data Files

One of your first tasks is to create a file of the information you wish to store and use. This "data base" may be created with your word processor (Scripsit, Lazy Writer, or Electric Pencil) or with DATA-WRITER's ENTRY program.

Data Base Organization

The data base of information must be organized according to strict rules so the program can recognize its individual elements. At the beginning of your data file, you need to put a list of these elements, called *fields*. Each field is identified by a label called a *field-label*. This is necessary so that DATA-WRITER can identify the fields for use and manipulation.

Each field-label must be put on a separate line and must be written in CAPITAL LETTERS. The maximum number of field-labels is 20. A double-period ("..") terminator must separate the end of the field-labels from the beginning of the first record. There may be no spaces in a field-label, so if you want to use multi-word labels, separate the words with another character, such as a hyphen or a plus.

While any field-labels will work, a simple name-and-address file might look like this:

```
NAME
PHONE
STREET
CITY+STATE+ZIP
..
```

Field-labels for a bibliography might look like this:

```
AUTHOR
TITLE
SUBJECT
DATE-OF-PUB
PUBLISHER
CITY-OF-PUB
REFERENCE
..
```

And field-labels for an inventory might look like this:

ITEM-NAME
INV#
STATUS
SIZE
BIN
UNIT-PRICE
STOCK-LEVEL
ORDER-LEVEL
..

Data in a Data Base

A *field,* described by a field-label, may contain any combination of numbers, letters, words or phrases. One set of information containing all your fields is called a *record.* Each record must contain the same fields as every other record in the file. And every field must be in the same position in every record.

When using your word processor, you may include up to 255 characters of data in each field. Spaces, or simply an <ENTER>, can be used to reserve space for a field when data for that field is not available.

A marker, called a *terminator,* must be placed after each record. The terminator tells the program where each record ends. DATA-WRITER uses a double period ("..") to terminate each record. Four periods ("....") are used to terminate the entire file. (The last record must have BOTH a record terminator and a file terminator.) *Each line in your file, including the terminator lines, must be ended with an* <ENTER>.

A short sample name-and-address data base might look like this:

```
NAME                    (List of
PHONE                     field-labels
STREET                    in CAPS)
CITY+STATE+ZIP          (Fill blanks)
..                      (Record terminator)
Nick Adams              (First record)
303-908-2481
23 Heming Way
Key West, FL 33040
..                      (Record terminator)
Cristoforo Colombo      (Second record)
                        (Missing data; space
One Nina Drive            reserved with <ENTER>)
New World, PA 14977
..                      (Record terminator)
N. Machiavelli          (Third record)
602-372-7391
12 Prince St
Venice, CA 90322
..                      (Record terminator)
....                    (File terminator)
```

Notice that, while there are upper and lower-case characters in the data base itself, *field-labels* may have only upper-case characters. In the above example, the blanks between words in the field-labels have been filled with pluses (though any printable character would do). Many features of DATA-WRITER will not work properly if spaces are contained in field-labels.

Items to remember are:

1. Field-labels must be listed at the beginning of every data base.
2. Field-labels must always be in capital letters.
3. There may be no blanks in field-labels.
4. No more than 20 fields are permitted in a record.
5. .. <ENTER> is a record terminator.
6. .... <ENTER> is a file terminator.
7. Every line in your data base *must* be terminated with <ENTER>. This includes the record and file terminators at the end of the file.

Special Considerations for Field-Labels

Although phrases are acceptable as field-labels, single words are better. You won't want to use a field-label that will limit the SORT or SELECT functions because it represents too much information.

An example of this is the use of NAME as a field-label. DATA-WRITER's SORT program can separate the last name from the rest of a name line, but for some uses this feature may not be as efficient as organizing the constituent parts into individual lines. Making use of the special extraction features of the sort program does, however, greatly facilitate the use of an existing list.

Let's look at the following simple data base:

NAME
STREET
CITY + STATE + ZIP
..
Mr. Richard King, Ph.D.
1483 Duke of Gloucester St.
Williamsburg, VA 23185
..
Dr. Kyle Kabbage, M.D.
16 W. Furrow Lane
Garden State, NJ 07625
..
....

You can sort by zip code by using the special extraction feature of the SORT program. You can also sort by last name (to alphabetize the list) by extracting it from the NAME line. However, you would not be able to insert just the last name in a form letter (for example, "Dear Dr. Kabbage") because last name is not in a field by itself.

You may gain considerable flexibility by creating a unique field for each individual data element. For example, the same list could be entered as:

TITLE
FIRST-NAME
LAST-NAME
DEGREE
STREET
CITY
STATE
ZIP
..
Mr.
Richard
King
Ph.D.
1483 Duke of Gloucester St.
Williamsburg
VA
23185
..
Dr.
Kyle
Kabbage
M.D.
16 W. Furrow Lane
Garden State
NJ
07625
..
....

When possible, fields should represent the smallest unit of information. You will then be able to *sort* and *select* by any key (field) and make complex select statements.

For example, you could sort by CITY. Or you could select if DEGREE = Ph.D. You could also select if ZIP > = 33023 AND ZIP < 38000. These functions will be described fully in the chapters relating to the individual programs.

When to Use Your Word Processor

There are some advantages to using your word processor to create and maintain your data base. And there are some advantages to using DATA-WRITER's ENTRY program.

The principal advantages to using your word processor is that DATA-WRITER will recognize and support *variable-length fields* up to 240 characters in length, but these can be created only with your word processor. In contrast, DATA-WRITER's ENTRY program can create fields no longer than 35 characters each (which may, however, be perfectly suitable for your application).

The principal advantages to using DATA-WRITER's ENTRY program, on the other hand, are its extensive validity checks, abbreviation table, and repeating fields, all of which make the data entry process much smoother and faster.

Adding Records to Your Data Base

You will often want to add names or information to an existing data base after it is completed and sorted. All you need to do is to enter it through your word processor or DATA-WRITER's ENTRY program.

There are two useful techniques if you use your word processor: One is to use the search function to find the proper place in the data base and use the insert function to make the entry. Similarly, the search function of your word processor will facilitate finding and correcting errors in a record.

The other technique is to enter new records at the end of the file; then run SORT to get them into the proper location. When adding several new records to the data base, it may be faster to use the second method. DATA-WRITER's ENTRY program uses the second method regardless of the number of new records being added.

Filenames and Saving a Data File to Disk

Each word processor has a slightly different filename structure. Files created with Electric Pencil will have an extension of "/PCL". In all cases, your filenames must follow standard TRSDOS filespec conventions.

When you save your data file to disk after you've created it with your word processor, it must be saved in ASCII format. When using Electric Pencil or Lazy Writer, this is done automatically. *For Scripsit you must specify the "A" option to save in ASCII format.* For example, to save a Scripsit file named "STORKS" on drive 1, type

S,A STORKS:1

Lazy Writer users should note that it is necessary to reverse the case before storing a Lazy Writer file on disk. This is accomplished by hitting the # symbol while in Edit mode.

Also keep in mind that DATA-WRITER or your word processor will write an output file to the first disk drive it finds that does not contain a write-protected diskette.

Reworking an Old File for DATA-WRITER Use

Some existing mailing lists and other data lists will work or can be edited to work with DATA-WRITER. The search and replace function of your word processor is the key to an easy conversion of these data files. If your list is stored in a sequential ASCII file, it probably has the same general format as a DATA-WRITER file. It is then practical to convert it for use. Some sophisticated users will be able to convert compressed files into ASCII files by writing software to do so.

Conclusion

An understanding of your own needs is the greatest asset you have for creating good data bases. DATA-WRITER makes it easy to implement and use them.

II. ENTRY: Adding Data to your Data Base

ENTRY is a program that allows you to add additional records to your data base. Although some users will want to use their word processor to maintain their data base, ENTRY will provide features not available with a word processor.

ENTRY can perform several types of data validity checks to reduce input errors to a minimum. In addition, ENTRY has features to increase the speed of the data entry process, including repeating fields and an abbreviaton look-up table.

The key to the ENTRY program is a *control file.* The control file contains information about the data base, including field-labels, validity checks, and other related information. The control file for a data base is always stored on the same disk drive as the data base itself. Control files use the extension "/CON". For example, if the name of a data base were DATA, then the control file name would be DATA/CON. After you have created a control file for a data base, it will automatically be used by the system whenever you use the ENTRY program.

To run the program, type "ENTRY" from DOS. The program will first ask you for the name of the data base. Type the name of the data base and press <ENTER>. The program assumes that the data base is on-line at this time. If you want to create a new data base, just type the name you want to give to it. (A filename must follow the TRSDOS format rules.)

If the data base file does not already exist, you will be asked to confirm that you want to create a new data base. When you use ENTRY to create a new data base, you must first create a control file. The program will prompt you through this process.

When creating a control file, you will need to enter information for each field in your data base. Most of this information is critical to the operation of the ENTRY program. For each field, you will need to enter:

1. Field Label
2. Field Length
3. Mandatory Entry or not
4. Type of Validity Check
5. Repeating Field or not

The *field-label* may be up to 20 characters in length. It may not contain spaces; however, it may contain special characters. Shorter field-labels are desirable to maintain an easy-to-read, uncluttered screen. Field-labels are discussed at length in Chapter I.

Next, you will be asked to enter the *field length*. This refers to the maximum length of the data to be entered in this field. DATA-WRITER will not allow you to enter information longer (having more characters) than what you specify for field length. The maximum field length allowed by the ENTRY program is 35 characters. If you need a longer field, you can use your word processor to create a data base with maximum field length of 240 characters, as discussed in Chapter I.

After entering the field length, the program will ask if the field is a *mandatory entry*. If it is mandatory, during the data entry process, your data entry operator will not be able to skip over this field. Some kind of entry must be made before the operator will be able to go to the next field.

Next, the program will ask you what type of *validity check* you want to perform on the data. You will have the following choices:

0 = No check
1 = All numeric
2 = Zip code check
3 = Length check
4 = Abbreviation expansion
5 = Dollar formatting

0 = No check

If you choose option 0, no check will be performed during data entry. (If the field is mandatory, the operator will still need to enter something into the field, even if no check is performed on what is entered.)

1 = All numeric

After the data have been entered in a field, each character entered will be checked to make sure it is either a number, a period or a minus sign. If any character does not meet these criteria, an error message will be generated, and the entry will have to be fixed.

2 = Zip code check

If you plan to use this feature, you must have specified a field length of *six* characters. This will accomodate all US and Canadian postal codes. The zip code check does not actually check for a valid zip; instead it checks for a plausible zip code. The US zip is five numeric characters. A Canadian postal code is six characters, alternating alpha and numeric. If a zip code fails both the US and Canadian checks, an error message will be generated, and the entry will have to be fixed.

3 = Length check

The data entered must use the entire field length specified for the field in the control file. If the operator enters information that is shorter than the field length, an error message will be displayed.

4 = Abbreviation expansion

This is not really a validity check; rather it is a way to speed up the data entry process. You will be able to enter an abbreviation, and the program will expand it to the full entry according to an abbreviation table you will set up after you have finished the field definition process.

Here's how it works: If you are working on a data base that includes several local communities, you might want to use an abbreviation expansion in the city field. During data entry, when you type "M", it will be expanded to "Minneapolis"... when you type "S", it will be expanded to "St. Paul"...

5 = Dollar formatting

Dollar formatting is not a validity check either. Any number entered will be reformatted to two decimal places. For example:

14 would be reformatted to 14.00
12.7 would be reformatted to 12.70

After you've entered the type of check, the program will ask if the entry program should *repeat the data* in that field from the previous record. If you choose a repeating field, then for each new record you enter into the data base, the default value for a repeating field will be the contents of the field from the previous record. This is often used when you will be entering the data in some sort of logical order. For example, if you were entering all your New York clients, you might want the CITY field to keep repeating New York until you decide to change it. Using this feature can greatly facilitate the data entry process.

After you have finished entering these criteria for one field, you will be given the opportunity to make changes. If you do not wish to make any changes, the program will request the same information for the next field. When you've defined all your fields, just type an asterisk " * " for the field label.

Finally, if you've specified any abbreviation expansions, the program will ask you for an abbreviation and an expansion. For the abbreviation, just type a one or two character code (for example, "M"). For the expansion, type the full expansion exactly as you want it to appear in the data base ("Minneapolis"). You may enter up to 100 abbreviations into your abbreviation look-up table. When you've finished, type an asterisk " * " for the abbreviation. The control file will automatically be written to disk, and you will then be able to add records to your data base.

Audible Signal

Whenever an invalid entry is attempted to a validated field, DATA-WRITER will generate a tone through the cassette port of your computer. If you attach a speaker-amplifier to your cassette port, your data entry operator will receive an audible signal of an invalid entry.

Data Entry

Entering data into the data base is quite easy. The field-labels will be displayed and the <UP-ARROW> and <DOWN-ARROW> keys may be used to move to any field you want. A graphic arrow will point to the current field. While you are entering a field, the cursor will move as in most word processors. The <LEFT-ARROW> and <RIGHT-ARROW> keys move the cursor.

Typing a character *overwrites* the current cursor position. A <SHIFT-DOWN-ARROW> *deletes* the current cursor charactor. A <SHIFT-UP-ARROW> puts the line editor in the *insert* mode. Any characters typed will be to the right of the cursor position, and all text to the right of the cursor will be pushed to the right. You will not be able to insert unless the last character of the field is not used for data. Pressing the <CLEAR> key will exit you from the insert mode. A <SHIFT-LEFT-ARROW> will delete the entire line up to the cursor, while a <SHIFT-RIGHT-ARROW> will delete from the cursor position to the end of the line.

After all the fields have been entered, you will be offered several options:

<CLEAR> will write the record to disk and prepare for the entry of the next record.

Typing <UP-ARROW> will allow you to re-edit the fields.

Typing <E> will end the entry position after the current record is written to disk.

Typing <A> will terminate the entry session *without* writing the current record to disk.

If you chose either E or A, the program will return to a menu:

1. Add more records to current data base
2. Add records to a different data base
3. Add to abbreviation table
4. End program

The menu is straightforward. The current data base is the one that you've just finished adding records to. Its name will be displayed at the top of the screen. If you want to add to the abbreviation table, the procedure will be as described above.

III. EDIT: Changing your Data

EDIT is a program that enables you to edit or review your data base without using a word processor. You will be able to correct errors made during data entry and also to keep the information in your data base current. EDIT is a word processor customized for editing data bases rather than text. It is similar, but not identical, to the text editors in LETTERS and REPORT, in response to specific needs of each program. Control codes are similar to those in Electric Pencil.

One especially useful feature of EDIT is that it will allow you to edit a data base of any size, limited only by the amount of disk space available. When a data base is too large to fit into memory, EDIT uses an input and output file so you can edit the data base in sections. Each section will typically contain 100 to 150 records of a name-and-address data base.

To run the program, type "EDIT" from DOS.

First, the program will request that you enter the name of your data base.

The memory buffer may be large enough to hold your entire data base. But some data bases will be too large to fit in memory at one time. If your data base is too large, you will have to edit the data base in sections. The screen would then display the message:

= = = > > > Memory full - Edit data base in sections

Next, the program will ask you to provide the name of the output file. This will be the name of the edited version of the data base. If you want to look at your data without making any changes (*"View Mode"*), press <ENTER> in response to this question. There will be no output file and EDIT will not save any changes you attempt to make.

If the data base is too large to be held in memory at one time, the output filename must be a different name if it is to be saved on the same diskette as the input file.

If the data base and the output file are on the same disk drive, the program will ask if you want to save the output file on the same diskette as the input file (Y/N). If you type "Y" for yes, the edited version of the data base will be written to the same diskette as the unedited version. If you type "N" for no, the program will prompt you when to insert the source and destination diskettes. The source disk refers to the disk with the data base before it is edited. The destination disk refers to the disk that will contain the edited version of the data base.

Caution: Take great care to insert the source or destination diskettes, as appropriate, when the prompt advises you to do so. It is your responsibility to insert the right diskette. Failure to follow instructions may destroy the data base! You can protect your source data base by placing a write-protect tab on it while you write your edited file to the destination diskette. If you do this, however, you must remove the tab before you can write anything more to the source diskette.

EDIT will now display the data base and is ready for editing. Each edit function is explained below and recapped in a summary at the end of this chapter.

The four *arrow keys* are used to move the cursor. You may scroll through the data base using the up and down arrow keys. Several control keys allow the fields to be edited. *Note:* <CONTROL> is accomplished by simultaneously holding down the <SHIFT> key and the <DOWN-ARROW>; so that, for example <CONTROL-C> requires pressing the <SHIFT>, <DOWN-ARROW> and <C> keys simultaneously. Typing <SHIFT-BREAK> on the Model I and <SHIFT-ZERO> on the Model III works like the shift lock on a typewriter.

The editor is normally in *overwrite* mode. This means that a character typed on the keyboard will replace the character the cursor is positioned over. To correct an erroneous character, move the cursor with the arrow keys to the position of the erroneous character, and type the new character. The old character will be replaced with the new character, and the cursor will advance one column to the right.

Display Current Column or Record Number

You can find the current column number of the cursor by pressing a <CONTROL-C>.

The current record number will be displayed by pressing a <CONTROL-V>. This feature will work regardless of where the cursor is in the record. When working on large data bases, there is typically a 3-5 second wait after pressing <CONTROL-V> before the current record number is displayed.

Insert Characters or Lines

Pressing a <CONTROL-F> will allow you to *insert characters* into the field where the cursor is positioned. The cursor will change size when <CONTROL-F> is pressed to let you know that you are now in the insert mode. EDIT will remain in the insert mode until another <CONTROL-F> or the <BREAK> key is pressed. All characters typed will be inserted ahead of the cursor. Typing a <CONTROL-G> will cause the program to enter the *insert line* mode. All characters after the cursor will be moved down one line and the cursor will change shape to advise you that you are in the insert line mode. If the cursor is in column one of a field when <CONTROL-G> is pressed, the program will insert a *new field* ahead of the current one. The insert line mode is terminated when the <ENTER> or <BREAK> key is pressed.

When using the insert line feature, take care to maintain the integrity of the data base. Make sure you do not inadvertantly change the number of fields in a record.

Delete Characters, Lines or Records

The <CONTROL-D> will *delete the character* the cursor is positioned over and may be depressed even while in the insert character mode.

Typing a <CONTROL-T> will *delete all characters* from the cursor position to the end of the current field. After typing a <CONTROL-T>, the cursor will be positioned over the carriage return symbol at the end of the field.

Typing a <CONTROL-Y> will *delete the entire field* regardless of the cursor position in that field. Again, be careful to maintain the integrity of your data base. Make sure you do not inadvertantly change the number of fields in a record. Typing a <CONTROL-P> will delete or *purge the entire record* regardless of the cursor position within that record.

Scrolling Through the File

You may scroll through the file manually with the up and down arrow keys. You may scroll through the file automatically with the <CONTROL-E> and <CONTROL-X> functions. Typing <CONTROL-E> will cause the program to begin scrolling to the end of the file, while typing a <CONTROL-X> will cause it to scroll to the beginning of the file. Pressing the <BREAK> key will terminate the scrolling function.

Typing a <CONTROL-B> will move the cursor immediately to the beginning of the file. A <CONTROL-N> will move the cursor to the end of the file.

Search for a Record Number or String of Characters

You can search for a particular record in your data base by typing a <CONTROL-S> (Search), providing that record is currently in memory. The program will ask which record is to be located. After you type the record number, the program will find the record and move the cursor to the first field of that record. When editing a data base in sections (because it is too large for the memory buffer), a search will not find a record that is not currently in the memory buffer.

You may also search for a particular string by typing a <CONTROL-L> (Locate). Like the <CONTROL-S> feature, the <CONTROL-L> will search only the current memory buffer. After typing <CONTROL-L>, the program will ask for the string to be located. Upper and lower case characters are different, so the search string must be typed correctly.

The program will then ask for a *replacement* string. If a replacement string is specified, all strings found during the search will be replaced by the replacement string. If no replacement string is specified, the program will stop when the first match string is found and will display the record on the screen. The cursor will be positioned over the first character of the string.

Typing a <CONTROL-R> (Resume) will continue a search and the program will look for the next occurrence of the string. This may be continued until no additional matches are found. A search may be resumed even after altering data.

Saving a Newly-Edited Data Base

When you have finished working with the current memory buffer, type a <CONTROL-O> (Oh) to save the contents of memory to disk. The cursor will automatically be moved to the beginning of the memory buffer before saving the file. If the entire file was able to fit into memory at one time, the program will end and return you to DOS Ready. If the file was too large to be contained in memory at one time, then EDIT will advise you to stand by for more data. Another section of the file will then be read into the memory buffer to allow you to edit it. This process will be continued until the entire data base has been read and written from the memory buffer. Each time a section of the data base is read, you must type a <CONTROL-O> to write the buffer to disk. When all disk operations are completed, the program end, and return you to DOS Ready.

If you do not wish to edit any more data, you may close the file and exit by typing a <CONTROL-ZERO>, If you type a <CONTROL-ZERO> by mistake instead of a <CONTROL-O>, the remaining input file will be written to the output file without offering you the option to edit it. You will then have to rerun EDIT and reload your data base.

Summary Of Control Key Functions

Press <CONTROL>, that is <SHIFT-DOWN-ARROW>, and

<C>	Display current column number.
<V>	Display current record number.
<F>	Enter the insert character mode. (Press <BREAK> to exit.)
<G>	Enter the insert line mode. (Press <BREAK> to exit.)
<D>	Delete the current cursor character.
<T>	Erase to the end of the current field.
<Y>	Delete the current field.
<P>	Delete (Purge) the entire record.
<B>	Move cursor to the beginning of file.
<N>	Move cursor to the end of file.
<E>	Scroll to the end of file. (Press <BREAK> to stop.)
<X>	Scroll to the beginning of file. (Press <BREAK> to stop.)
<S>	Search for record number to be specified.
<L>	Locate a string to be specified.
<R>	Resume search for a string.
<O>	Write the current memory buffer to disk.
<ZERO>	Write the remainder of the input file to disk.

Other Special Functions

Press:

<RIGHT-ARROW>	Move cursor one position to the right.
<LEFT-ARROW>	Move cursor one position to the left.
<SHIFT-RIGHT-ARROW>	Tab to the next tab stop to the right.
<SHIFT-LEFT-ARROW>	Tab back to the previous tab stop.
<SHIFT-BREAK>	(Model I only.) Same as shift lock on a typewriter.
<SHIFT-ZERO>	(Model III only.) Same as shift lock on a typewriter.
<CLEAR>	Move cursor to beginning of current record.

IV. MANAGE: Manipulating Your Data Base

MANAGE is a program that gives you the capability to restructure your data base. You will be able to delete fields, create new fields, reorder the fields, merge data bases, and split a data base into two smaller ones. Generally, the MANAGE program will help you adjust your data base for unanticipated changes in format and content.

To run the program, type "MANAGE" from DOS. The program will load, and you will be given three choices:

1. Field manipulation
2. Merge data bases
3. Split data base

Field Manipulation

The ability to manipulate fields is quite useful, and is one of DATA-WRITER's most powerful features. You will often find yourself wanting to add new fields to your data base or to delete old ones. Sometimes you'll want to merge two fields into one field to make room for new information. The field manipulation feature of MANAGE is easy to use.

After choosing option 1, you will be asked to enter the name of your data base. MANAGE assumes that the data base is on-line at this time. Next, the program will ask you for the name of an output file. This is the name that you will give to your new, manipulated data base. MANAGE will check the disk to make sure there is enough space for the new file to be written before continuing.

The program will now display all your field labels. You can:

I = Insert a new field
D = Delete a field
S = Swap the positions of two fields
M = Move a field to a different position
A = Append one field to another
R = Restore format of input file
E = End

There is a specific syntax requirement associated with each of the field manipulation commands. To perform any of these commands, just press the appropriate key. Then you will enter additional information about your command. The syntax is quite simple and is described below.

Insert

To insert a new field, the program needs two items of information: What position should the new field be in, and what will be the field-label for the new field? Enter the information in the format "#,LABEL". For example, if you want to insert a new field "COST" into position five, you would first press the <I> key, then type:

5,COST <ENTER>

The new field COST will be inserted into position five, and all the following fields will be moved down one position to make room. It is not possible to add a new field if all twenty fields are already filled.

Delete

Deleting a field is accomplished by pressing the <D> key and then typing the number of the field you want to delete. For example, to delete your field "COST" from position five, you would first press the <D> key, then type:

5 <ENTER>

Swap

To swap the positions of two fields, first press the <S> key. Then type the numbers of the two fields you want to swap in the format #,#. For example, if you wanted to swap fields 3 and 5, you would press the <S> key and then type:

3,5 <ENTER>

Move

If you want to move a field from one position to another, first press the <M> key and then type the number of the field you want to move and the position you want to move it to. If you wanted to move the fifth field to the first position, you would type:

5,1 <ENTER>

Append

Sometimes you may want to combine two fields. You may append one field to another by first pressing the <A> key and then typing the numbers of the fields to be combined. First type the field to be appended, then type the field to which it will be appended. For example, if you wanted to append field four (say, STATE) to the end of field three (say, CITY), you would type:

4,3 <ENTER>

When using the append command, the field-labels will be combined as well as the contents of the fields. On the screen, an appended field will be displayed as two fields joined by a graphic block. In the output file, the new field-label for the combined field will be its previous label, a PLUS SIGN and the name of the appended field tagged on the end. For example,

CITY+STATE

You may not *append* a newly inserted field nor one that has already been appended.

Restore

If you have started manipulating fields but want to restore them to their original condition, just press the <R> key. The fields will be restored to their original order, and you will be able to start over again or end.

When you have finished manipulating fields, press the <E> key to indicate that you want to end manipulating. If you have created a new field, the program will ask you to enter the field contents. What you enter here will appear in this field *in every record* of your data base. Just pressing <ENTER> will put a blank field in every record.

If you have appended two fields together, you will be asked to type the characters you want to separate the data contents of each record. For example, if you have appended the STATE field to the CITY field, you will probably want a comma and a space between the two fields in every record; for example, "Minneapolis, Minnesota". So you would enter

<,> <SPACE> <ENTER>

Next the program will perform the data manipulation. The input file will be read and the new output file will be written. When the data manipulation is complete, the program will tell you how many records were read and written.

Merge Data Bases

Merging two data bases is a common procedure. For example, let's say you run a business where people call in to inquire about your service. You want to send each inquirer a form letter which you will keep as part of your permanent records so you can follow up in the future. An easy way to do this is to use two separate data bases. One will be called "DAYFILE" and the other "PERMFILE". Each day you receive inquiries and enter inquirers' names into DAYFILE. At the end of the day, you run your form letter for all the records in DAYFILE. Then merge DAYFILE and PERMFILE to produce an updated data base with all the records.

After selecting the merge option, the program will ask you for the name of the first data base. Next, you will be asked for the name of the second data base. MANAGE will then check to make sure the field-labels from the first data base match the field-labels for the second data base. *You may not merge two data bases with different field-labels.* MANAGE will then ask for the name of your new output file. This is the merged file that will contain all the records from both data bases.

MANAGE will display the field-labels and ask you to enter the label for the "merge field". This is the field that the program will use to sort the records as they are merged. If both data bases are in the same sorted order, then a merge on the "merge field" will produce an output file in sorted order. If the input files are not in any particular sorted order, or if only one of them is, then just press <ENTER> when asked for the merge field. If you are merging on a particular field, you will be asked if the field contains alpha or numeric data.

The two input files will be read and a new output file will be written. The program will tell you the number of records in each input file and the number of records that were written to the output file.

Split a Data Base

It is possible that, as your data base grows, you will want to split it into two smaller sections. If you choose to split a data base, you will first be asked for the name of the data base to be split.

Next, you will be asked for the name of the first output file. This file-name must contain a drive number. Then you will be asked for the file-name of the second output file. This must also include a drive number. If both output files are going on the same drive, MANAGE will ask if you will want to swap diskettes.

Finally, MANAGE will ask how large you want the first output file to be. If you type a number that is less than 1000, then MANAGE will place that many records in the first output file. If you type a number greater than 1000, MANAGE will interpret it as the number of characters to be placed in the first output file. For example, if you typed "10000", MANAGE would create the first output file, write up to 10,000 characters to it and then write the rest of the data base to the second output file. (A record would not be split in the middle, however.) At the completion of a split, MANAGE will tell how many records were placed in each output file.

* * *

The MANAGE program is very powerful. You will be able to control your data base easily. One word of caution is in order. If you restructure your data base, the *control file* used for the data entry program will no longer be accurate. It will be necessary to kill the old control file "FILENAME/CON" and then to use ENTRY to recreate the control file. *Do not* attempt to use an old control file on a restructured data base.

V. STATS: How to Verify Your Data Base

After you have assembled your data base, it is not necessary to proofread every record just to check for format errors. While each program has its own error checking and reporting functions, the STATS program provides a means to report format errors prior to the actual printing of letters, reports or labels.

In addition to its ability to check for format errors, STATS also reports vital information you may need for you to use other programs effectively. STATS can also be used to print or display the entire data base.

To run the program type "STATS" from DOS.

You will first be requested to enter the name of your data base. After you enter your data base name, the program will ask you to

Select Type of Listing:

1. Print all records
2. Print errors only

Which ?

Then the program will ask you to

Select output :

1. Screen
2. Printer
3. Disk File

Which ?

If you choose screen output, the program will list all information on your screen. You may use the <S> key to pause (start and stop) scrolling the screen.

If you choose printer output, the program will send all output to the parallel printer and will assume that the printer is on-line and ready. If the system seems to hang, check to make sure your printer is on and ready.

If you choose disk file output, the program will ask you for the name of the disk file. In this case, an ASCII print image file will be stored under the name you specify.

When you run STATS for errors only, you will get a report that looks like this:

```
ERROR LISTING OF FILE - CLIENT/PCL          PAGE 1

= = > >  Record 5 - Missing information in record
         TITLE          ____  Mr.
         FIRST-NAME     ____  Anderson
         LAST-NAME      ____  Beaverton Inc.
         COMPANY        ____  333 Weslaur Rd.
         STREET         ____  Primrose
         CITY           ____  CA
         STATE          ____  99108
         ZIP            ____  ..

= = > >  Record 96 - Record has too many fields of
         information
         TITLE          ____  Dr.
         FIRST-NAME     ____  John
         LAST-NAME      ____  Whinton
         COMPANY        ____  California Computer Data
         STREET         ____  Attn: Purchasing Dept.
         CITY           ____  1487 Palm Drive Court
         STATE          ____  Ferndale
         ZIP            ____  CA
         ?              ____  95033

* * ERROR LISTING COMPLETED * *

NUMBER OF VALID RECORDS ON FILE = 367
NUMBER OF INVALID RECORDS ON FILE = 2
TOTAL = 369

* * Field Labels and Maximum Field Lengths * *

         TITLE          ____  6
         FIRST-NAME     ____  9
         LAST-NAME      ____  12
         COMPANY        ____  28
         STREET         ____  30
         CITY           ____  27
         STATE          ____  2
         ZIP            ____  7

= = = > > > RUN COMPLETED
```

In Record 5, the FIRST-NAME was missing, and all other fields were shifted up one, pulling the *terminator* into the place for the ZIP field. Thus the error message, "Missing information in record." In Record 96, the addition of the extra field "Attn: Purchasing Dept." forced the ZIP field to over-run the terminator and trigger the error message, of "Record has too many fields of information."

Armed with the printout of errors, you can now use the "Search" function of DATA-WRITER's EDIT program or of your word processor to edit your data base to bring the errant records into line.

The maximum *field length* information is needed when you are designing a REPORT. This information tells you how wide to make the columns on the report to accommodate all the data.

STATS is useful to verify the integrity of your data base. You should always run STATS after you have entered data using your word processor.

VI. MATH: Using Math Procedures

MATH is a sophisticated program that allows you to calculate fields of your data base. You may enter equations, freely mixing field-labels and numeric constants. Addition, subtraction, multiplication, division and exponentiation can be used in your calculations. Each math procedure may utilize up to twenty equations. In addition, there are ten *scratch pad* (temporary storage) locations where you can store intermediate results.

To run the program, type "MATH" from DOS. You will first be requested to enter the name of your data base. Your data base must already contain all the fields that will be calculated by the MATH program (even if a field is blank). You cannot create new fields with the MATH program. You can only calculate values to put into existing fields. If your data base does not contain all the fields you will need, then run the MANAGE program to create new fields before running MATH.

After you enter the name of your data base, MATH will check the field labels. Then the program will ask you to enter the name of the output file. This is the name of the new data base which will contain the results of your mathematical calculations. After you type the name, MATH will check the disk space to make sure there is enough room to store the output file. Both the input and output files will need to be on-line during the math procedure.

Next, MATH will ask you to enter the name of the *math procedure* file. Just press <ENTER> to create a new math procedure file. The math procedure file is simply a set of equations that you want performed on your data base. The MATH program allows you to enter up to twenty equations. If you plan to do the same calculations on a regular basis, then you will probably want to save your equations on disk. You might be using DATA-WRITER to invoice clients. Here's an example: Let's say one of the fields in your data base is PARTS and another is LABOR. SALES-TAX and TOTAL-DUE are fields that will be calculated. The equations might be:

SALES-TAX = .05 * PARTS
TOTAL-DUE = SALES-TAX + PARTS + LABOR

Since you will be sending out invoices each month, you might want to save the equations so you won't have to type them at the end of each month.

There are several rules for entering equations:

1. Equations are entered one at a time and checked for validity before being entered into the system.

2. Equations may contain field-labels, numbers and operators. There must be *one space* between each of the parts of the equation.

3. The field-labels T0, T1, T2 ... T9 may be included as temporary storage locations to hold intermediate results of your formulas.

4. Five numeric operators may be used: + for addition, - for subtraction, * for multiplication, / for division, and * * for exponentiation.

5. Operations are evaluated from first equation to last equation and from left to right.

6. Parentheses are not allowed.

7. Only one field-label (or T#) may be on the left of the equal sign.

Some samples of valid equations are:

```
SALARY = HOURS * 9.35
TOTAL-DUE = CURRENT + 30-DAYS + 60-DAYS
  + 90-DAYS
RADIUS = DIAMETER / 2
AREA = RADIUS * * 2 * 3.14159
VOLUME = LENGTH * WIDTH * HEIGHT
T3 = .7 + T2 - VALUE * FACTOR
SQRT = VALUE * * .5
```

Some examples of invalid equations are:

SUBTOTAL + VALUE = NEW-VALUE	(Only one field label may be on the left of the =)
ANSWER = RESULT + 5	(Missing a space before the = sign)
TOTAL = (.05 * AMOUNT) + AMOUNT	(Parentheses not allowed)

If you make a mistake in typing an equation, the program will tell you the kind of mistake and let you re-enter the equation. When you have finished entering equations, you will be given the opportunity to make any changes you want. Then you will be asked if you want to save the equations on disk as a math procedure file. If you answer "Y" for yes, you will be asked for a file name for the procedure file, and it will then be saved on disk. This file may be used at a later time to re-perform the same calculations.

Finally, the input file will be read, the math equations will be performed, and the output file will be written. Your input file (the original data base) will not be altered in any way. The output file will contain the results of the calculations. At the end of the procedure, the program will display a short summary of the procedure, including any errors that occured.

MATH is a very powerful program. You will be surprised at how easy it is to perform complex arithmetic calculations on your data base.

VII. SELECT: Making a Subset of your Data Base

SELECT is a program that lets you isolate carefully chosen records out of a larger data base and create a data file of the selected information.

With this subset data base, you may then send targeted form letters, write control reports, identify and order low stock items from an inventory list, or type special labels for just the female membership of a group, and so on.

To run the program, type "SELECT" from DOS. After the program loads, it will ask for the name of your data base that you are going to select from. Then it will ask you to provide a new filename for the selected subset.

The program then prompts:

SELECT IF

Your response will be statements that will instruct the program to look at each record, one at a time, for specific characteristics. If any of your criteria are met, the software will trap that record and store it on disk, along with its fellows, under the new filename. In our hypothetical SELECT run, let's answer with a simple statement first and see the results.

We will ask the computer to select those persons in a mailing list who live in California. The most direct way to do this is to respond:

SELECT IF STATE = CA

Notice that there is one space between each of the parts of the select statement. "STATE" must be one of the field-labels in our data base. The computer will print the words "SELECT IF." You need to type only "STATE = CA".

Another way to accomplish the same goal is to define the area you want within its zip code limits. The resulting select statement is:

SELECT IF ZIP > 89999 AND ZIP < = 96199

In this example, all zip codes greater than 89999 and less than or equal to 96199 will meet the SELECT-IF criterion. This SELECT-IF statement, therefore, chooses those addresses whose zip codes are in California.

Now suppose we have an inventory and want to create a report of only low-level stock items. We would use two of the fields in our data base, ORDER-LEVEL and QUANTITY.

SELECT IF QUANTITY < = ORDER LEVEL

This statement says to select any records where the quantity is less than or equal to ORDER-LEVEL.

You may use nine equivalency relationships in forming your select statements. They are:

=	Equal to
<	Less than
< =	Less than or equal to
>	Greater than
> =	Greater than or equal to
< >	Unequal to
> >	Equal to the right part of the string
< <	Equal to the left part of the string
> <	Equal to any part of the string (instring)

In addition, you may freely use the words "AND" and "OR" to form complex select statements. The only limitation for each select statement is that it may not exceed 255 characters. For example, let's say we wanted to mail to three targeted neighborhoods from our mailing list data base. ZIP is one of the *field-labels* in our data base.

SELECT IF ZIP = 55423 OR ZIP = 55425 OR ZIP = 55429

If the zip code in a record is any one of the three, the record will be selected, otherwise, it will be skipped. The "OR" relationship says: "select this record if any one of the conditions is true."

Another way to select these same zip codes is to make three separate select statements. That is:

SELECT IF ZIP = 55423
SELECT IF ZIP = 55425
SELECT IF ZIP = 55429

In this case, because the criteria are on separate lines, it's as if they were linked by an "OR" statement. If these statements were to be run, they would result in the program looking at each record in a data base and asking if the first criteria (ZIP = 55423) has been met. If the record was not selected, then the same record would be scrutinized for a match to the parameters of the second statement (ZIP = 55425). If the record was not selected by the second statement, it would be checked for a match with the third statement. Only when it had failed the test for ALL the select statements would it be rejected. You may define as many as 30 separate select statements having three elements (words and relational operators) each, or somewhat fewer more complex select statements.

The AND statement allows you to select records if all your select criteria are met. For example, let's say you want to select from your data base all the females in Michigan. In other words, you want to select records if the SEX field is female *and* if the STATE field is Michigan. If both criteria are met, then the program will select the record. If either criterion is not met, then we want to skip the record. You would type:

SELECT IF SEX = F AND STATE = MI

Right String, Left String, Instring:

Suppose we want to select from our data base all the people who are doctors. But suppose we do not have a separate field for M.D. in our data base. Instead we have just typed it at the end of the name field; as "Fred Smith, M.D." In this example we would want to select all records where the right-most characters of the name field are "M.D." Our select statement would be:

SELECT IF NAME >> M.D.

Suppose we wanted to select all the records of people whose first name was "Fred." In our data base, first name and last name are in the same field. We would type:

SELECT IF NAME << Fred

Notice that we're selecting first name from the left side of the field. Also notice that "Fred" is in upper and lower case letters. If we had a record in the data base with a first name of FRED (all caps), it would not be selected, since upper and lower case are considered to be different by the SELECT program. This could easily be overcome by using the OR in a select statement:

SELECT IF NAME << Fred OR NAME << FRED

Suppose you're a stock broker, and you're using DATA-WRITER to keep track of your clients' investments. One of your fields is a list of codes that tells you what stocks a client has. For example, a typical field might look like this: IBM ATT GM CDC Now we want to select all clients who have investments in GM. Our select statement would be:

SELECT IF INVESTMENT >< GM

This would select all records where GM was contained anywhere in the INVESTMENT field. Or, if we wanted to select all clients that had investments in GM or IBM:

SELECT IF INVESTMENT >< GM OR INVESTMENT >< IBM

Or, if we wanted to select all clients that had investments in both GM and IBM:

SELECT IF INVESTMENT >< GM AND INVESTMENT >< IBM

In the above examples, we always selected one word that contained no spaces. What if we wanted to select all the records where the CITY field is Los Angeles. Since Los Angeles contains a space, we must tell the SELECT program that it should be interpreted as one word. We would get a syntax error if we typed

SELECT IF CITY = Los Angeles (Incorrect)

Instead, we use two slashes "//" to indicate the start and end of the field contents. The correct way to make this SELECT statement is:

SELECT IF CITY = //**Los Angeles**// (Correct)

A complex SELECT statement may be made using the "//" as needed. For example:

SELECT IF CITY = //**New York**// **OR CITY** = **Washington**

When using the SELECT program, the new data base of selected records will be in the same sorted sequence as the original file. If your original list is in last-name alphabetical order, then your new file will also be in last-name alphabetical order (although it will probably contain fewer records than the original file).

The best way to get the feel for this remarkable program is to try it with your data base. After you have built whatever list you need, try creating several different types of sub-lists.

VIII. SORT: Ordering your Data Base

SORT is a fast data manipulation program. Its purpose is to "re-shuffle" the records of your data base into a new order for some specific purpose. The program provides the capability of a two-level sort (a sort within a sort). Each level may be sorted in alphabetical or numerical sequence and in ascending or descending order. The maximum number of records that can be sorted by DATA-WRITER should not prove to be a constraint to most users. DATA-WRITER can sort a data base of up to 4500 records.

SORT has three additional features that allow you greater flexibility in structuring and using your data base:

1. Extract a *zip code* from a line. You may select the zip code as a sort KEY even if it is embedded in a "CITY + STATE + ZIP" field.
2. Extract a *last name* from a line. You can separate the last name from a line of "FIRST-MIDDLE-LAST-NAME" with a high degree of accuracy due to special software that examines a name field and separates out the last name.
3. Reformat a *date* from MM/DD/YY format to YYMMDD format. This will allow you to sort by a date field.

To run the program, type "SORT" from DOS. SORT will first ask you to

Enter the name of the data base to be sorted

When you answer with a data base that is on-line, the program will load the field-labels from your data base and list them on the screen. The program will then ask you to

Enter the field-label of the primary sort key

Type the field-label you wish to use to re-order your file. "Primary" refers to the most significant sort field when you are doing a two-level sort. For example, let's say you want to sort your list alaphabetically by COMPANY and then sort it alphabetically by LAST-NAME within COMPANY. Your primary sort key will be COMPANY; your secondary key will be LAST-NAME. If you are not doing a two-level sort, just enter the field-label you will be using to sort the file.

When you enter your field-label, type it exactly as it appears. If, for example, you want to sort by zip code, but it is embedded in a line called "CITY + STATE + ZIP", you would type the field name "CITY + STATE + ZIP", and not just "ZIP". The program will ask:

Special sort options:

1. **Extract and sort by Last Name**
2. **Extract and sort by Zip Code**
3. **Reformat date**
4. **Standard sort - No special options**

Which ?

If you want to sort by last name, and the last name is not a separate field, you may select option 1 and extract the last name from a larger chunk of information. For example, in each of the following fields, "Jones" would be extracted as the last name to be used in an alphabetic sort:

Alice Jones
Mr. John Jones, Manager
Dr. F. Jones, Jr.
Mr. Carl Jones III

If you were choosing to extract the zip code from a field that contained City, State and Zip, then select option 2 because you want to extract the zip code from a larger piece of information.

If the last name or the zip code is in its own field then *do not* use the special extraction options — use the standard sort.

If the field you want to sort by is a date in MM/DD/YY format, then you'll want to choose special SORT option 3. You must use this option to sort any date field that is not in YYMMDD format (no slashes). For example, when you're entering records in your data base on April 1, 1982, you could enter 820401. Then you could perform a sort on this field without choosing special option 3. If however, you entered the date as 4/1/82 or 04/01/82, you must choose option 3 and the date will automatically be converted to YYMMDD format for the purpose of the sort. (It will not be changed in the data base itself).

The program will ask you if you want a numerical sort or a alphabetical sort. A general rule is to use a numerical sort on any field that is quantified or measured (that is, one that represents a quantity, such as dollar amounts, inventory levels, etc.). However, *do not* use a numerical sort on any field that contains commas, dollar symbols, slashes or any other special character. Use an alaphabetical sort.

Zip code sorts should generally be alphabetical sorts. This will accomodate Canadian, foreign and nine-digit zip formats. If you know that all the records in your data base have five-digit U.S. zip codes then you may sort in numerical sequence.

The program then asks you to

Enter the sort order A = Ascending D = Descending

Ascending means A to Z, or 1 to 10. Descending means Z to A, or 10 to 1. After you type your response, the program will ask you the same information for the secondary sort. If you do not need a two-level sort, then just press <ENTER> when the program asks you for the field-label of the secondary sort.

Finally, the program will ask you to

Enter the name of the output file

Type a new filename under which to store the newly sorted data base. If you use Electric Pencil to access the new data base, be sure to add the file extension "/PCL" to your filename. Also remember that, if you do not specify the drive number, the program will write the newly sorted file to the first disk drive that does not contain a write-protected diskette.

The filename might look like this:

ZIPSORT/PCL:1

The program will check the amount of disk space available on the drive you specified for the output file. If there is not sufficient disk space to perform the sort, the program will not begin the sort, and it will ask you to re-enter a new filename for the output file.

The best way to explore the capabilities of this program is to do just that. Run SORT using a sample data base. Try several of the options and features. You may be surprised how "smart" the program is and how many different, useful data bases you can create from one original.

IX. LABELS: Creating Mailing Labels

LABELS is a program that allows you to print mailing labels, one to four across, on your printer. You will be able to create a label format file and save it on disk so you can reuse it for future printings. The format of the labels is completely adjustable. LABELS also allows you to chain data bases during the printing process.

To run the program, type "LABELS" from DOS.

First, you will be asked to enter the filename(s) of the data base(s) to be printed. Use either a space or a comma to separate the names from each other. If you want to print two sets of labels for a data base, just type its name twice. An example would be:

NAMES1,NAMES2,NAMES2

Here, one set of labels would be printed for the first data base, and two sets of labels would be printed for the second. When chaining files, the field-labels for all the data files specified must be the same. In other words, you will only be able to chain files if the formats of the data bases are identical. The program expects that the first data base specified is on-line at this time.

If you have specified that more than one data base is to be printed, the program will ask if you want to stop for disk mounts. If all the data bases specified can fit on-line, then it is not necessary to stop for disk mounts. If you will need to swap diskettes, then you will want the program to stop for disk mounts.

Next, the program will ask you to enter the name of the file containing the label format. This is a file that tells the program how the labels should be printed. If you have previously stored a label format file on disk, just type its file-name, and it will be loaded in from disk. If you don't have a label format file stored on disk, then just press <ENTER> to create a new one.

You will be asked several questions when you create a new label format file.

First, you will be asked how many labels you want to print across the paper. You may print one to four labels across the paper.

Next, you will be asked how many lines there are from the top of one row of labels to the top of the next row. Stock labels generally run 5, 6 or 7 lines. The number you enter here should include the blank lines between the labels.

The program will then ask for the width of the print zone on a label. This refers to the maximum number of characters that will be printed on a line of each label. The minimum label width is five characters. A typical label size is 30 or 32 characters. If a field is larger than the print zone on a label, the right-most part of the field will be truncated during the printing. You can check your maximum field length using STATS.

If you are printing more than one label across the paper, you will asked to enter the number of blank spaces between labels. There must be at least one blank space between columns of labels.

The program will now ask for the left margin. This refers to the column on the printer where the labels will start to be printed. The left margin may be between 1 and 50.

Next, you will be asked to enter a "preset string". The preset string will allow you to print a common line on all labels. For example, if you want to print labels where the top line is always "Marketing Manager", then you would type "Marketing Manager" for the preset string.

After entering all the above information, you will be given the opportunity to make changes to it. Finally, all field-labels will be displayed, and you will be asked to select the fields you want to print on each line. For example, let's say we have a data base with the following fields:

FIRST-NAME
LAST-NAME
STREET
CITY
STATE
ZIP
plus a preset string of "President."

We might type a fourline label format like this:

PRESET
FIRST-NAME LAST-NAME
STREET
CITY, STATE ZIP

Notice how the field-labels are used to create a formatted label. You may type spaces and commas between the field-labels on a given row. During the printing, the spaces and commas will be included. The keyword PRESET is interpreted just like a field-label.

You have now completed creating your label format, and you will probably want to save it on disk so you can just load in this format at a future time. LABELS will ask you for the filename to save your label format file. Then the label format file will be saved on disk.

Next, you will be asked if you want a printer test. It is generally good practice to run a test before you begin printing a large data base. It will give you an opportunity to check out the printer and adjust the paper before beginning. After a printer test you may test the printer again, start printing the labels, or re-edit the label format. You can alternate between edit and test until you are satisfied with the format and then may begin printing the labels.

Printing may be stopped at any time by holding down the <S> key. You will be able to restart the printing from the current record, a different record, or from a different data base. You also have the option to end the program. At the completion of the program, a summary report will be displayed on the screen telling you how many labels were printed.

X. LETTERS: How to Create a Form Letter

LETTERS is a program that lets you create and run form letters, envelopes and mailing labels. You can create a form letter with your word processor, or you can use the *Text Editor* in this program. This chapter explains how to create your form letter using either the LETTERS text editor or your word processor and then how to print it.

You can use this program with little preparation, though planning the format and content of your letter will aid you in producing predictable results the first time.

Planning your letter

LETTERS has a capacity of 6000 characters, which will accomodate a form letter of about three pages.

To plan your document, you will need to know which fields of your data base you wish to insert into the form letter. If you are using your word processor to create the form letter, it will be necessary to get that information from the STATS program. If you are using the LETTERS Text Editor, however, you will be able to display the field-labels on the screen by using a <CONTROL-L>. *Note:* <CONTROL> is accomplished by simultaneously holding down the <SHIFT> key and the <DOWN-ARROW>.

This is what a listing of the FIELD-LABELS and their lengths from the STATS printout might look like:

TITLE	3
FIRST-NAME	9
LAST-NAME	12
COMPANY	28
STREET	30
CITY	27
STATE	2
ZIP	5

Identifiers and Terminators

When creating your form letter, each *field-label* or *keyword* must be enclosed within *identifiers*. The purpose of the identifier is to tell the computer that it is not to print what it sees, but to use it to find information. The identifier is *a parenthesis and an asterisk*, with *no spaces*, at each end of the field-label, like this:

(*FIRST-NAME*)

In addition to the field-labels in your data base, the keyword (*REC*) may be used to display the record number. Every form letter needs a *terminator* at the end. This takes the form of four periods immediately followed by a carriage return (“....<ENTER>”). This is the same as the file terminator that ends your data base and serves a similar purpose.

You can construct a simple form letter using the identifiers and field-labels as in the next example. Notice there is a direct substitution of *field-label* for each word or phrase wanted.

(*TITLE*) (*FIRST-NAME*) (*LAST-NAME*)
(*STREET*)
(*CITY*), (*STATE*) (*ZIP*)

Dear (*TITLE*) (*LAST-NAME*):
On January 1, 1982 we will be increasing our prices by 12.6% across the board. Orders received prior to that time, regardless of the projected shipping or billing dates, will be honored at the pre-1982 prices.

Sincerely,

Winston J. Gillespie
General Manager
....

The printed letter would look something like this:

Dr. Allen Little
414 Drurey Lane
Minneapolis, Minnesota 55429

Dear Dr. Little:
On January 1, 1982 we will be increasing our prices by 12.6% across the board. Orders received prior to that time, regardless of the projected shipping or billing dates, will be honored at the pre-1982 prices.

Sincerely,

Winston J. Gillespie
General Manager

Insert from Keyboard

Now let's add a function for adding the date using the keyboard insertion feature. The format for this is (*?PHRASE*). For example,

(*?Enter the date*)

Notice the question mark in the line and the phrase "Enter the date". When the program is run, it will print the letter, including insertion requests, up to the point where it contains the *keyboard insert request* (*?Enter the date*). Then it will stop and the screen will display the prompt: "Enter the date".

After you enter the date into the computer, the program will ask you,

Should this data be used for all letters (Y/N) ?

If your answer is "Y" for yes, that value will become fixed for the remainder of the run, but may be changed or cleared by pressing a <CONTROL-K> while in Text Editor Mode.

Each time you run the program using this form letter, a request for new data will be made. You can elect to use the same value for all the letters of that run, or change it for each letter. You can thus update the letter as you go, change it when you stop the program, or change it only when you re-run the letter.

You are not limited to the date or, for that matter, just to one keyboard input. Up to ten keyboard inputs may be inserted during a run of the program.

If for some reason you want to stop the program, for instance, to use the editing controls or to re-format the margins, all the keyboard inputs will remain intact unless you clear them in Text Editor Mode with a <CONTROL-K>. If you clear them, you will need to re-enter them when you restart the printing.

Now our form letter will look like this:

```
(*TITLE*) (*FIRST-NAME*) (*LAST-NAME*)
(*STREET*)
(*CITY*), (*STATE*) (*ZIP*)

(*?Enter the date*)
Dear (*TITLE*) (*LAST-NAME*):

On January 1, 1982 we will be increasing our prices by
12.6% across the board. Orders received prior to that
time, regardless of the projected shipping or billing
dates, will be honored at the pre-1982 prices.

Sincerely,

Winston J. Gillespie
General Manager
....
```

And the printout would look like this:

Dr. Allen Little
414 Drurey Lane
Minneapolis, Minnesota 55429

December 26, 1981

Dear Dr. Little:

On January 1, 1982 we will be increasing our prices by 12.6% across the board. Orders received prior to that time, regardless of the projected shipping or billing dates, will be understood to be completed and will be honored at the pre-1982 prices.

Sincerely,

Winston J. Gillespie
General Manager

Format Codes

Now lets wrap up the letter by using *format codes* and by printing envelopes.
You can control the "shape" of the letter by typing *format lines.* They contain *format codes* that are read and acted upon by the computer but not printed. They must contain a leading period which the computer recognizes as a signal that says "don't print me...just do what I say." format codes must be the first and only characters on the line and must be terminated with an <ENTER.

These codes are the following (replace "?" with a number):

.LL? Sets the line length to "?". The computer defaults to .LL64

.LM? Sets left margin to "?" spaces. The computer defaults to .LM6

.LS? Sets the line spacing to "?". The computer defaults to .LS1

.PL? Sets the length of the page to "?". The computer defaults to .PL66

.TM? Sets the number of blank lines at the top of of the page before printing begins. The computer defaults to .TM5

.BM? Sets the number of blank lines that will be placed at the bottom of the page. The computer defaults to .BM5

.CM Reserves a comment line. You can put notes after this that will be displayed, but not printed (up to the next carriage return).

.IY? Sets a special *indenting* feature that can indent both the left and the right margins by an additional "?" number of spaces to "set-off" a line or lines.

.IN Turns off the "IY" feature, returning the settings to those previously used by the program.

.SP Sets a *stop* at the end of each page to allow for a change of paper when using manually-fed single sheets. Pressing <ENTER> restarts the printing on the next page or at the beginning of the next letter.

.FF Forces a *form feed* prior to the end of page. In effect, it ends a page and calls for a new one. *Do not use .FF to end a form letter.*

.PT? Sets the pitch for a NEC 5530 printer to "?" (10 or 12).

Envelopes

Many people who use a mailing list program prefer to print each envelope at the time they print each letter. They then stack them in herringbone fashion for stuffing and sealing. In order to accomplish this, a call for the name and address is inserted twice and the *format codes* are used to do the work.

Here's an example of a form letter with an envelope printing:

```
.CM This is a sample form letter. Creation date: 11/4/81
.SP
.LM6
.LS1
(*TITLE*) (*FIRST-NAME*) (*LAST-NAME*)
(*STREET*) (*CITY*), (*STATE*) (*ZIP*)

(*?Enter the date*)

Dear (*TITLE*) (*LAST NAME*):

On January 1, 1982 we will be increasing our prices by
12.6% across the board. Orders received prior to that
time, regardless of the projected shipping or billing
dates, will be honored at the pre-1982 prices.

Sincerely,

Winston J. Gillespie
General Manager
.FF
.LM40
.LS2
(*TITLE*) (*FIRST-NAME*) (*LAST-NAME*)
(*STREET*)
(*CITY*), (*STATE*) (*ZIP*)
....
```

The ".CM" and the text that follows is a comment line and will not be printed. It may be used to store the document creation date or any other information you feel is important.

The ".LM6" at the top of the letter resets the left margin to "6" in the event the program was stopped prior to a complete envelope printing.

The ".SP" tells the computer to stop when it reaches the end of a page or encounters an ".FF" form feed in order for the operator to change paper. In this example, ".SP" stops the program to let the operator insert an envelope.

The use of the ".FF" in this case tells the computer where to stop to wait for you to insert an envelope. Pressing <ENTER> will restart the printing again. A ".FF" should not be used at the end of the letter, since there is an automatic feed at the end of each letter. In multi-page letters an automatic form feed is inserted when the top margin plus the bottom margin plus the lines printed equals the page length.

The ".LM40" sets the left margin in from the edge of the paper stop by 40 spaces, putting the address in the proper place, in this example, for a #10 envelope. Printing begins at the 41st column on the printer. The ".LS2" sets the line spaces to two for the name and address on the envelope. Some printers will require that you add one to the line spacing you desire. For double spacing, you might have to use ".LS3" rather than ".LS2". Experimentation will determine if this is necessary with your printer.

The repeat of the *field-labels* tells the computer what to print on the envelope, and the "...." marks the end of the form letter (including the envelope).

One-Up Labels

For one-up continuous-feed labels or for printing envelopes all at once, use the appropriate form letter format.

A format for one-up labels with seven spaces from the top of one label to the top of the next might be like this:

```
.TM0
.BM0
.PL7
(*FIRST-NAME*) (*LAST-NAME*)
(*STREET*)
(*CITY*), (*STATE*) (*ZIP*)
....
```

The reservation of only seven lines per "page" made the listing simple. A variety of numbers of lines could be used without changing the other codes. This "form letter" has only field labels and control codes. Single envelopes could be run with the addition of the ".SP" and ".LM40". Like any other "letter," the format for typing labels can be saved as a file and re-used at a future time. (Of course, you can also use DATA-WRITER's LABELS program for printing one-up labels.)

Special Printer Control Codes

DATA-WRITER also allows you to send special codes to your printer. Many printers accept control codes to do things such as emphasized printing, compressed printing, subscripts, and superscripts. You can easily send your printer any code or codes. Simply insert into your form letter the keyword (*###*), where ### is the appropriate decimal number that controls your printer. For example, to set emphasized printing on the MX-80 printer requires a two character sequence: "Escape E." The MX-80 uses a decimal code 27 as "Escape". To send the sequence to the printer you would include in the text of the form letter,

```
(*27*)E
```

Since an E is the same as an ASCII 69, we could accomplish the same thing by sending

```
(*27*)(*69*)
```

Notice that there are no spaces between the parts of the control sequence. You could also use this facility to send a bell (*7*) to your printer to mark the end of each form letter or to signal an operator to insert an envelope.

Running the Program

Now that you have planned your form letter, you're ready to run the program. Type "LETTERS" from DOS.

LETTERS will first ask you to enter the name of your database.

Type in the name of your names-and-addresses data base or other appropriate file. Next it will request that you

> **Enter the name of the form letter**
> **or press <ENTER> to create a new one:**

If you type the filename of a form letter, it will be loaded from disk, and the *menu* will be displayed. If you have not yet created and stored a form letter, press <ENTER> to go to Text Editor Mode.

Text Editor

LETTERS will display the following instruction:

> **USE <CONTROL–O> TO RETURN FROM EDIT MODE.**
> **Press <ENTER> to begin edit/review.**

After you press <ENTER>, the program will be ready for text editing. Each edit function is explained below and recapped in a summary at the end of this chapter.

As in most word processors, the four *arrow keys* are used to move the cursor. You can scroll up or down, or move right or left, by pressing the appropriate key. Several control keys allow your letter to be created and edited. *Note:* <CONTROL> is accomplished by simultaneously holding down the <SHIFT> key and the <DOWN-ARROW>. Typing <SHIFT-BREAK> on the Model I and <SHIFT-ZERO> on the Model III works like the shift lock on a typewriter.

The editor is normally in *overwrite* mode. This means that a character typed on the keyboard will replace the character the cursor is positioned over. To correct an erroneous character, move the cursor with the arrow keys to the position of the erroneous character, and type the new character. The old character will be replaced with the new character, and the cursor will advance one column to the right.

Display Field-Labels

You can display all field-labels in your data base by pressing <CONTROL-L>. Pressing <ENTER> returns you to edit mode.

Insert Characters or Lines

Pressing a <CONTROL-F> will allow you to *insert characters* into the field where the cursor is positioned. The cursor will change size when <CONTROL-F> is pressed to let you know that you are now in the insert mode. The editor will remain in the insert mode until another <CONTROL-F> or the <BREAK> key is pressed. All characters typed will be inserted ahead of the cursor.

Typing a <CONTROL-G> will cause the program to enter the *insert line* mode. All characters after the cursor will be moved down one line and the cursor will change shape to advise you that you are in the insert line mode. If the cursor is at the first column of a line, a new line will be inserted in front of the one the cursor is on.

Delete Characters, Lines or Records

The <CONTROL-D> will *delete the character* the cursor is positioned over and may be depressed even while in the insert character mode.

Typing a <CONTROL-T> will *delete all characters* from the cursor position to the end of the paragraph. After typing a <CONTROL-T>, the cursor will be positioned over the carriage return symbol at the end of the paragraph.

Typing a <CONTROL-Y> will *delete the entire paragraph* regardless of the cursor position in that paragraph.

Scrolling through the Letter

You may scroll through the letter manually with the up and down arrow keys. You may scroll through the letter automatically with the <CONTROL-E> and <CONTROL-X> functions. Typing <CONTROL-E> will cause the program to begin scrolling to the end of the letter, while typing a <CONTROL-X> will cause it to scroll to the beginning of the letter. Pressing the <BREAK> key will terminate the scrolling function.

Typing a <CONTROL-B> will move the cursor immediately to the beginning of the file. A <CONTROL-N> will move the cursor to the end of the file.

Clear Keyboard

Insert Values If you wish to change a previously entered *keyboard insert value* which you are using for all letters, just press <CONTROL-K>. You will then be given the opportunity to change keyboard insert data during printing. In fact, you will have to re-enter them all after pressing <CONTROL-K>.

Exit

When you have finished creating or editing your letter, press <CONTROL-O> (Oh) to leave Text Editor Mode.

Saving to Disk

When you exit Text Editor Mode, the program will ask if you want to save the letter to disk. If you answer with a "Y" for yes, the program will then ask if you want to use the current filename or a new one. If you choose to use the current filename, the program will erase the existing file and write over it. If you elect to use a new filename, the program will ask for its name and then create a new file using the new filespec. If you use Electric Pencil, be sure to add the extension "/PCL" to the filename.

The default filename for a form letter is LETTER/TXT, but more than likely you will want to use your own system of names. The program uses the same conventions as TRSDOS for naming files.

After saving the file to disk, the program displays the menu:

<<< PRINTING NOT YET STARTED >>>

You have the following options:

1. **Begin printing with record #1**
2. **Begin printing with a different record #**
3. **Review and/or change form controls**
4. **Enter edit mode to review and/or change letter**
5. **Quit and close files**

Your Selection ?

Form Controls

If you were to choose 3, the form controls and their default values would be displayed. They look like this:

*** * FORM CONTROL DEFAULT VALUES * ***

1. Page length	=	**66**
2. Top margin	=	**5**
3. Bottom margin	=	**5**
4. Line spacing	=	**1**
5. Left margin	=	**6**
6. Line length	=	**64**

Enter the number to change or press <ENTER> to return.

The values displayed are default values. The LETTERS program will use them if you do not elect to change them or if your letter does not use format codes that change them.

To change any default value, select its line number. The program then asks you to input a new value. When you respond with a new value, that value is then displayed in the position of the one it is replacing.

If you press <ENTER> instead of choosing a value, you will return to the menu.

Selection 4 will take you back to Text Editor Mode, so you can modify your form letter. *(See Text Editor above.)*

Printing your Letter

To begin printing, select either option 1 or 2 from the menu. If you choose option 1, the program will begin printing with the first record. If you select option 2, the program will ask you for the record number to begin printing. (The *field-labels* at the beginning of your data file are considered to be record #0.)

If you have included any *keyboard insert requests* in your letter, printing will stop when it reaches each one so you can enter your input. Then the program will ask if you want to use the same value for all letters. *(See the discussion above in this chapter under the heading, "Insert from Keyboard.")*

Pressing the <S> key during the printing process will stop the printing. This would be necessary in the event of a printer jam, discovery of an error in the letter, or some other reason.

After pressing <S>, you will be at the menu again. The program will display the sequence number of the record that was being printed. You have the same options as before. If you choose to continue, the program will reprint from the *beginning* of the letter. When you begin printing again, be sure to re-position the paper at the top of the page.

After all records in the data base have been printed, the program will display any errors that occurred during printing. Some of the errors may have prevented the printing of a particular record, and it will be of interest to you to know where they occurred. A missing field or delimiter is the type of error that will prevent a record from being printed.

Blank Fields

Normally, LETTERS will skip over a field insertion when the field contents are blank. For example, if the inside address on a letter includes the field COMPANY, as in

```
(*NAME*)
(*COMPANY*)
(*STREET*)
(*CITY*), (*STATE*) (*ZIP*)
```

but the record did not happen to have a company name, then the program would ignore the company line altogether, rather than print a blank line.

There is a way, however, to force LETTERS to print blank lines for missing fields. After you type "LETTERS" from DOS, hold down any key until the program has completed loading. Then release the key. Now when you print a letter, LETTERS will include a blank line on the printer for each missing field.

Summary of Control Key Functions

Press <CONTROL>, that is <SHIFT-DOWN-ARROW> and

<F>	Enter the insert character mode. (Press <BREAK> to exit.)
<G>	Enter the insert line mode. (Press <BREAK> to exit.)
<D>	Delete the current cursor character.
<T>	Erase to end of paragraph.
<Y>	Erase entire paragraph.
<B>	Move cursor to beginning of file.
<N>	Move cursor to end of file.
<E>	Scroll to end of text. (Press <BREAK> to stop.)
<X>	Scroll to beginning of text. (Press <BREAK> to stop.)
<O>	Exit the editing mode.
<L>	Display field-labels.
<K>	Clear keyboard insert values.

Other Special Functions Press:

<RIGHT-ARROW>	Move cursor one position to the right.
<LEFT-ARROW>	Move cursor one position to the left.
<SHIFT-RIGHT-ARROW>	Tab to the next tab stop to the right.
<SHIFT-LEFT-ARROW>	Tab back to the pevious tab stop.
<SHIFT-BREAK>	(Model I only.) Same as shift lock on a typewriter.
<SHIFT-ZERO>	(Model III only.) Same as shift lock on a typewriter.

XI. REPORT: How to Create a Report

You can use the REPORT program to create a printed report of information from any of your data files. You can use your word processor to generate a report format. But you will probably find it easier to use the *Text Editor* of the REPORT program because it has special features to facilitate the design of your report.

In this chapter we will explain the format controls, show examples of where they apply, and then build a sample report. As in creating a form letter, planning the format and contents of your report will make its production speedier and predictable.

Planning your Report

To create statements that will print data beneath appropriate column headings, you need to know how long the data fields are. If the statements allow for too few positions, the data will be truncated and some essential information may be lost.

At the end of the STATS printout is a listing of all of the field-labels and the maximum number of characters in each of the fields. The following list is an example of what field-labels might look like for a name-and-address file:

```
* * FIELD LABELS AND MAXIMUM FIELD LENGTHS * *

    NAME                  ____ 28
    ADDRESS               ____ 14
    CITY+STATE+ZIP        ____ 22
```

General Format of a Report

There are four major sections to every report file: the *Parameter Section, Heading Section, Format Section* and *Subtotal Section*. The general format is displayed below:

PARAMETER SECTION

.... (terminator)

HEADING SECTION
.... (terminator)

FORMAT SECTION
.... (terminator)

SUBTOTAL SECTION
.... (terminator)

Parameter Section

The *Parameter Section* consists of imbedded commands similar to those used in the LETTERS program. These commands are the format codes that control the printing of the text. They determine the shape of the page; that is, they tell the program what limits have been set for printing.

There are seven format commands, as follows (enter a number for ("?"):

.PL? Sets the total *page length,* including top and bottom margin to "?" (defaults to 66 lines).

.TM? Sets the number of blank lines used for a *top margin* to "?" (defaults to 5 lines).

.BM? Sets the number of blank lines used for a *bottom margin* to "?" (defaults to 5 lines).

.SP *Stops printing* at the end of a page to allow the operator to remove and insert single sheet forms. (Default is automatic page advance.)

.PT? Sets the *pitch* on a NEC 5530 printer to "?"; enter 10 or 12.

.PC? Special *printer code*. Send an ASCII code to your printer. Enter any appropriate number for "?".

.FF Forces a *form feed* after a subtotal break; that is, it ends a page and calls for a new one.

A sample parameter section for a report might look like this:

```
.PL66
.TM4
.BM2
.SP
....      (terminator)
```

We put in a ".PL66" for a page length of 66 lines; we set the top margin for four blank lines with ".TM4"; we set the bottom margin to two blank lines with ".BM2"; and finally we instructed the printer to stop at the end of each printed page for a change of paper with a code ".SP". Also, notice the "...." terminator that marks the end of the *Parameter Section*.

A "...." terminator ends each section. This one tells the computer that the program is done defining the page parameters. If you do not need any special parameters, you must still type "...." for an end-of-section symbol. If you don't, the program will indicate that an error has occurred when it checks the format. The program will use the default values if no parameters are included.

The Heading Section

The Heading Section describes what will be printed at the top of each page on your report. A sample heading might look like this:

```
        ACCOUNTS RECEIVABLE FOR 10/28/81  PAGE (#)
-------------------------------------------------------
NAME   ADDRESS   CITY   ZIP   CURRENT    +30 +60 +90
-------------------------------------------------------
....
```

The Heading Section follows the Parameter Section. In this example we have included a title and date, an automatic page numbering code, column headings, and two decorative lines. Notice the "...." terminator that marks the end of the section.

The title and date are obvious. They tell the reader what the report is about. Location control for these inputs is entirely manual. There is, however, an aid available to help you put in the spaces needed to center or otherwise position these heading items: <CONTROL-C>. When the cursor is over a character, you may press the <CONTROL-C>, and the program will display the column position of the cursor (indicated as the number of characters from the beginning of the line).

Another feature that can be of help when you want to place the cursor in a specific column is <CONTROL-P>. When pressed, the program will ask you what column you want to place the cursor in. After typing a number, the cursor will automatically tab to the column you specify.

Another feature allows you to print the page number on each page of your report. Just insert "(#)" in the Heading Section. A consecutive number will be printed at that position on each page of your report, starting with the first page.

A decorative or separating line, or some other series of characters, is often desired in the Heading Section of a report. The auto-repeat function, using <CONTROL-R>, allows you to insert a pre-determined number of characters into a line. After typing <CONTROL-R>, the program will ask you the number of times to be repeated. Then it will ask you for the string to be repeated. For example, if you wanted an 80-character line, you would type <CONTROL-R>, then "80 <ENTER>" and then "-<ENTER>". The text would then contain 80 columns of minus signs or dashes.

Column headings tell the reader the significance of the contents of each vertical column. They should, therefore, line up over their respective data. The positioning of these inputs is manual; that is, you determine visually where the column headings should go and put them there. Because most reports will be wider than the TRS-80's 64 character screen, use <CONTROL-C>, to tell you where you are and <CONTROL-P> to get you to a specific column position.

Format Section (where the data will be printed out)

The *Format Section* describes how the columns and rows of data in the report will be printed. A sample Format Section might look like this:

L1-30(*NAME*) R35-48(*PHONE*) D55-65(*AMOUNT-DUE*)

....

Here's what all this means:

"L1-30(*NAME*)" is called a *field-position* statement because it tells the program how and where to position a data field on the line. There cannot be any blanks between the elements in a field-position statement. However, there must be at least one blank between each individual field-position statement.

There are three essential elements in a field-position statement.

L, R or D = *Left, Right* or *Decimal*. It determines on which side of the reserved column the field will be justified. You can cause the columns of your report to print block left, block right or, when working with dollar amounts, to line up the decimal points. Normally, L, R or D would be capital letters. However, if you want a total at the end of the report or subtotals at some breakpoint, then use a lower case *l, r* or *d* for that field.

number-number = Beginning and ending column numbers of a data field. This tells the program where to begin printing a data field and where to end it. If your data are longer than the field you specify, the program will truncate the data (chop off the characters furthest to the right). In the example, "number-number" was "1-30", where 1 is the first column in the row and 30 is the 30th. It means "begin printing in column one and stop printing in column 30."

(*FIELD = -LABEL*)	The identifiers. They tell the program to "look inside here and see what field to print." Also, the special field-label "REC" will print the record number of the current record.

When creating or editing a report file, you can move the cursor between the heading section and the format section in order to check the position of a column heading with the aid of <CONTROL-C>. Just use the arrow keys to move the cursor.

A Sample Report

The example shown next for *field-position statements* allows room for the maximum size of the fields shown in the field-label printout above. A space will be placed between each column.

```
R1-28(*NAME*) R30-44(*ADDRESS*)
L46-68(*CITY+STATE+ZIP*)
```

You may have up to 20 columns of information and up to five rows for each record in your data base. For example, the following Format Section will print three lines per record.

```
Blank line terminated with <ENTER>
L1-27(*NAME*) L29-41(*PHONE*) <ENTER>
L1-27(*STREET*) L29-70(*CITY+STATE+ZIP*)
<ENTER>
....(terminator)
```

It will print one blank line and then two lines of information for each record in your data base. The print-out sample that follows is an example:

Joyce James 2125 Ulysses Av. N.	612-332-7839 Mpls., MN 55417
N. Machiavelli 12 Prince Street	714-372-7391 Venice, CA 90322
Geoffrey Chaucer 1340 Canterbury Lane	303-916-4296 Kent, OH 44240
Henry Anjou 1154 Kings Way	402-374-1829 Beckett, NJ 07090

The layout of the columns and headings is the most difficult task. There is a difference in appearance between actual spacing in the heading and virtual positioning in the field-position statement. You may find it helpful to print an entire row of consecutive numbers across the printout paper you will be using. This will provide a template for layout purposes. Armed with this and a printout of STATS, you should be able to plan a creditable document in very little time.

Subtotal Section

When you want your report to include subtotals, simply type the name of the field that you want to use to break the report into subtotals. For example, let's say you are keeping track of your company's expenditures with DATA-WRITER. Each check written has been entered with an AMOUNT and CATEGORY, among other fields. When you print your report, you'll probably want a subtotal of expenditures for each CATEGORY as well as the grand total.

Before printing the report, you will have sorted your file by CATEGORY using the SORT program. On your report, you want a subtotal in the AMOUNT column every time the CATEGORY field changes. In the SUBTOTAL section, you type the field-label that will control when the program will print subtotals. In our example, we would type CATEGORY, since we want subtotals to be printed whenever the CATEGORY field changes. Follow this entry with a "...." terminator.

```
CATEGORY
....
```

We must also have included a lower case *l, r* or *d* for AMOUNT (probably a "*d*") as part of the field-position statement. Subtotals and totals will be printed for every field that used a lower case *l, r* or *d* as part of the field-position statement.

Note: Even if you don't want any subtotals, it is still necessary to place the "...." at the end of the Subtotal Section.

Putting It All Together

To show a completed report format, we will use an inventory report format and data base.

```
.PL66
.TM4
.BM2
.SP
....
                    INVENTORY FOR THE CHANDLERY

ITEM    INV#    STATUS    SIZE CLASS    UNIT-PRICE
STOCK-LEVEL ORDER-LEVEL <ENTER>
_____________________________________________
_____________________<ENTER>

....
R1-10(*ITEM*) L11-13(*INV#*) L15-28(*STATUS*)
L30-33(*SIZE*) L35-39(*CLASS*) D42-51(*UNIT-
PRICE*)
L54-65(*STOCK-LEVEL*) L68-78(*ORDER-LEVEL*)
<ENTER>
....
....
```

Note: The headline or title is not centered on a screen that is only 64 characters wide but would center on a report that is 80 characters wide.

The lines for the column headings "wrap over" into the next line, but they are not terminated by an <ENTER>. That means the program thinks of them as being on one line that goes beyond the end of the screen and wraps them around for your viewing convenience.

The characters for the decorative separator also and field-position statements also wrap around to the next line. This means the program thinks of them as a continuous line. If terminated by <ENTER>, however, they will not wrap around.

It is possible to start over in column #1 for as many as five separate lines.

NOW...lets run it!

To run REPORT

Type "REPORT" from DOS. First the program will ask for the filename of your data base. After you type the name of your data base, the program will then ask you to enter the name of your report (format) file. If you already have a report stored on disk, type in its filename. If you want to create a new report format, just press <ENTER>, and the program will enter the Text Editor Mode.

Text Editor

The text editor used to create (or edit) a report file is similar to the text editor used in the form letter program.

REPORT will display the following instruction:

> **USE <CONTROL-O> TO RETURN FROM EDIT MODE.**
> **Press <ENTER> to begin edit/review.**

After you press <ENTER> the program will be ready for text editing. Each edit function is explained below and recapped in a summary at the end of this chapter.

As in most word processors, the four *arrow keys* are used to move the cursor. You can scroll up or down, or move right or left, by pressing the appropriate key. Several control keys allow your letter to be created and edited. *Note:* <CONTROL> is accomplished by simultaneously holding down the <SHIFT> key and the <DOWN-ARROW> and pressing the next key specified. Typing <SHIFT-BREAK> on the Model I and <SHIFT-ZERO> on the Model III works likes the shift lock on a typewriter.

The editor is normally in *overwrite* mode. This means that a character typed on the keyboard will replace the character the cursor is positioned over. To correct an erroneous character, move the cursor with the arrow keys to the position of the erroneous character, and type the new character. The old character will be replaced with the new character, and the cursor will advance one column to the right.

Display Field-Labels

You can display all field-labels in your data base by pressing <CONTROL-L>. Pressing <ENTER> returns you to edit mode.

Insert Characters or Lines

Pressing a <CONTROL-F> will allow you to *insert characters* into the field where the cursor is positioned. The cursor will change size when <CONTROL-F> is pressed to let you know that you are now in the insert mode. The Editor will remain in the insert mode until another <CONTROL-F> or the <BREAK> key is pressed. All characters typed will be inserted ahead of the cursor.

Typing a <CONTROL-G> will cause the program to enter the *insert line* mode. All characters after the cursor will be moved down one line and the cursor will change shape to advise you that you are in the insert line mode. If the cursor is at the first column of a line, a new line will be inserted in front of the one the cursor is on. The insert line mode is terminated automatically when the <ENTER> is pressed or may be terminated by pressing the <BREAK> key.

Delete Characters, Lines or Records

The <CONTROL-D> will *delete the character* the cursor is positioned over and may be depressed even while in the insert character mode.

Typing a <CONTROL-T> will *delete all characters* from the cursor position to the next carriage return. After typing a <CONTROL-T>, the cursor will be positioned over the carriage return symbol at the end of the line.

Typing a <CONTROL-Y> will *delete the entire line* from the previous carriage return to the next carriage return, regardless of the cursor position in that line.

Scrolling through the Report

You may scroll through the report manually with the up and down arrow keys. You may scroll through the report automatically with the <CONTROL-E> and <CONTROL-X> functions. Typing <CONTROL-E> will cause the program to begin scrolling to the end of the report, while typing <CONTROL-X> will cause it to scroll to the beginning of the report. Pressing the <BREAK> key will terminate the scrolling function.

Typing a <CONTROL-B> will move the cursor immediately to the beginning of the file. A <CONTROL-N> will move the cursor to the end of the file.

Display Column Number of Cursor

Pressing <CONTROL-C> will display the cursor's column number in a line. This function helps you line up headings and data in your report.

Remember that one line includes all characters and spaces up to the next carriage return. The display of column number lasts a few seconds, then disappears.

Position the Cursor at Column Number

Pressing <CONTROL-P> enables you to instruct the program to move the cursor to the position of your choice on its current line. When you press <CONTROL-P>, the program will instruct you to

Place the cursor in column number:

Respond by typing a column number, then <ENTER>. The program returns to edit mode with the cursor at the column position you specified.

Repeat a Character

This function enables you to draw decorative or separating lines across your report. When you press <CONTROL-R>, the program will ask you to input the number of times you want a character or string of characters to appear in succession. Respond with the number of times.

Then the program will ask you to enter the string that is to be repeated. Type the character on string you want inserted.

To draw a line on an 80-column printer, you might use 80 "+" or "-" or "=", or 40 "< >".

Exit

When you have finished creating or editing your letter, press <CONTROL-O> (Oh) to leave Text Editor Mode.

Saving to Disk

When you exit Text Editor Mode, the program will ask if you want to save the report to disk. If you answer with a "Y" for yes, the program will then ask if you want to use the current filename or a new one. If you choose to use the current filename, the program will erase the existing file and write over it. If you elect to use a new filename, the program will ask for its name and then create a new file using the new filespec.

After saving the file, the program displays the menu:

<<< PRINTING NOT STARTED YET >>>

You have the following options:

1. **Begin printing with page #1**
2. **Begin printing with a different page**
3. **Review and/or change form controls**
4. **Enter edit mode to review and/or change report**
5. **Quit and close files**

Your selection ?

Form Controls

If you were to choose 3, the form controls and their default values would be displayed. They look like this:

*** *FORM CONTROL DEFAULT VALUES* ***

1.	**Page length**	=	**66**
2.	**Top margin**	=	**5**
3.	**Bottom margin**	=	**5**
4.	**Page advance**	=	**Automatic**

Enter the number to change or press <ENTER> to return.

The values displayed are default values. The REPORT program will use them if you do not elect to change them or if your report does not use format codes to change them.

To change any default value, select its line number. The program then asks you to input a new value. When you respond with a new value, that value is then displayed in the position of the one it is replacing.

If you press <ENTER> instead of choosing a value, you will return to the menu.

Selection 4 will take you back to Text Editor Mode, so you can modify your report. (*See Text Editor above.*)

Printing your Report

To begin printing, select either option 1 or 2 from the menu. If you choose selection 1, the program will begin printing with the first page of your report. If you select option 2, the program will ask you for the page number to begin printing.

Pressing the <S> key during the printing process will stop the printing and return you to the menu.

Summary of Control Key Functions

Press <CONTROL>, that is <SHIFT-DOWN-ARROW>, and

<F>	Enter the insert character mode. (Press <BREAK> to exit.)
<G>	Enter the insert line mode. (Press <BREAK> to exit.)
<D>	Delete the current cursor character.
<T>	Erase to end of line.
<Y>	Erase entire line.
<B>	Move cursor to beginning of file.
<N>	Move cursor to end of file.
<E>	Scroll to end of text. (Press <BREAK> to stop.)
<X>	Scroll to beginning of text. (Press <BREAK> to stop.)
<O>	Exit the editor and return to main program.
<L>	Display the field-labels.
<C>	Display the column number of the cursor in the line.
<P>	Move the cursor to a specified position in the line.
<R>	Repeat the characters to be input.

Other Special Functions

Press:

<RIGHT-ARROW>	Move cursor one position to the right.
<LEFT-ARROW>	Move cursor one position to the left.
<SHIFT-RIGHT-ARROW>	Tab to the next tab stop to the right.
<SHIFT-LEFT-ARROW>	Tab back to the previous tab stop.
<SHIFT-BREAK>	(Model I only.) Same as shift lock on a typewriter.
<SHIFT-ZERO>	(Model III only.) Same as shift lock on a typewriter.

TECHNICAL SUPPORT: What to do in case of a system failure

DATA-WRITER is a sophisticated software system that requires strict syntax rules for your data base, form letters and reports. Ignoring these rules will surely impede your use of DATA-WRITER and may even result in a system failure.

In case of a failure, check to make sure you have done the following things correctly:

1. Did you save your data base, form letter or report in ASCII format? Remember, Scripsit requires you to use the "A" option when saving your files. If STATS gives you an "Input Past End" error message, this is most likely the problem. Also, if a form letter or report loads improperly, it probably was not saved in ASCII format.

2. Does the last record in your data base have a ".." terminator followed by <ENTER>? Also, is the last line in your data base "...." followed by <ENTER>? If SORT gives you a "Disk Full" error, check these two items.

3. Are the field-labels all upper-case letters? If LETTERS and REPORTS gives you "Undefined Field Label" messages for all your insertions, check the field-labels in your data base. Also remember to change case before storing your data base when using Lazy Writer.

4. If the LETTERS program prints garbage at the end of your form letters, check to make sure the letter ends with "...." and <ENTER>.

5. There is approximately 6K of memory available for a form letter. This is equivalent to about five pages of text. If your form letter will not load, check to make sure it does not exceed this length.

While it is possible that a bug or two has nosed its way into DATA-WRITER, we ask that you re-read the User's Manual before you contact us for help. If you still need assistance, please write to us, describing the problem in detail and how it can be duplicated. We will respond promptly.

Software Options, Inc.
P.O. Box 970
Bowling Green Station
New York, N.Y. 10274

LIMITED WARRANTY

This program is provided "as is." Software Options warrants the diskette on which the program is furnished to be free from defects in materials and workmanship under normal use for a period of ninety days from the date of delivery to you as evidenced by a copy of your receipt.

THIS IS THE ONLY WARRANTY WE EXTEND.

SOFTWARE OPTIONS DOES NOT EXTEND AND SPECIFICALLY DISCLAIMS ALL OTHER WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU.

Software Options' entire liability and your exclusive remedy shall be the replacement of any diskette not meeting Software Options' Limited Warranty and which is returned to Software Options with a copy of your receipt.

In no event will Software Options be liable to you for the damages, including incidental or consequential damages, arising out of the use of or inability to use this program.

Data-Writer™

REGISTRATION FORM

Please fill out and return this registration form to us so we can keep you apprised of upgrades, fixes, applications, and other interesting happenings.

NAME _____

ADDRESS _____

_____

CITY _____

STATE _____ ZIP _____

PURCHASE DATE _____

PURCHASED FROM _____

Send this completed form to

Software Options, Inc.
P.O. Box 970
Bowling Green Station
New York, N.Y. 10274

SOFTWARE OPTIONS INC™

NEW DATA-WRITER UPDATE PROGRAM

A major enhancement to Data-Writer will be available on November 1st. While EDIT is a good general purpose editing program, it is often difficult to use for changing specific field data. Now we have UPDATE, a program that will allow the user to change data in his data base in the same format as ENTRY.

But UPDATE does more.

First, it will build a key file, using any key you specify, including imbedded last name or zip code.

Second, you can search by key or by record number, forward or backward, through your data base for each occurrence of the search string you specify. Access to any record in your file, EVEN ON DISK, takes about one second! You can edit the record in ENTRY format (with field-labels on the screen), then write your changes to a new file.

Third, UPDATE will merge the changes into your original data base.

DATA-WRITER will be sold complete with the new UPDATE program after November 1st at $145. Registered owners of Data-Writer may purchase UPDATE separately for $20 plus shipping and diskette.

To order, mail us a check for

1. UPDATE program: $20.
2. A blank diskette or $3 diskette charge.
3. NY State sales tax (if applicable).
4. Shipping charge: $2 US and Cda; $5 foreign.

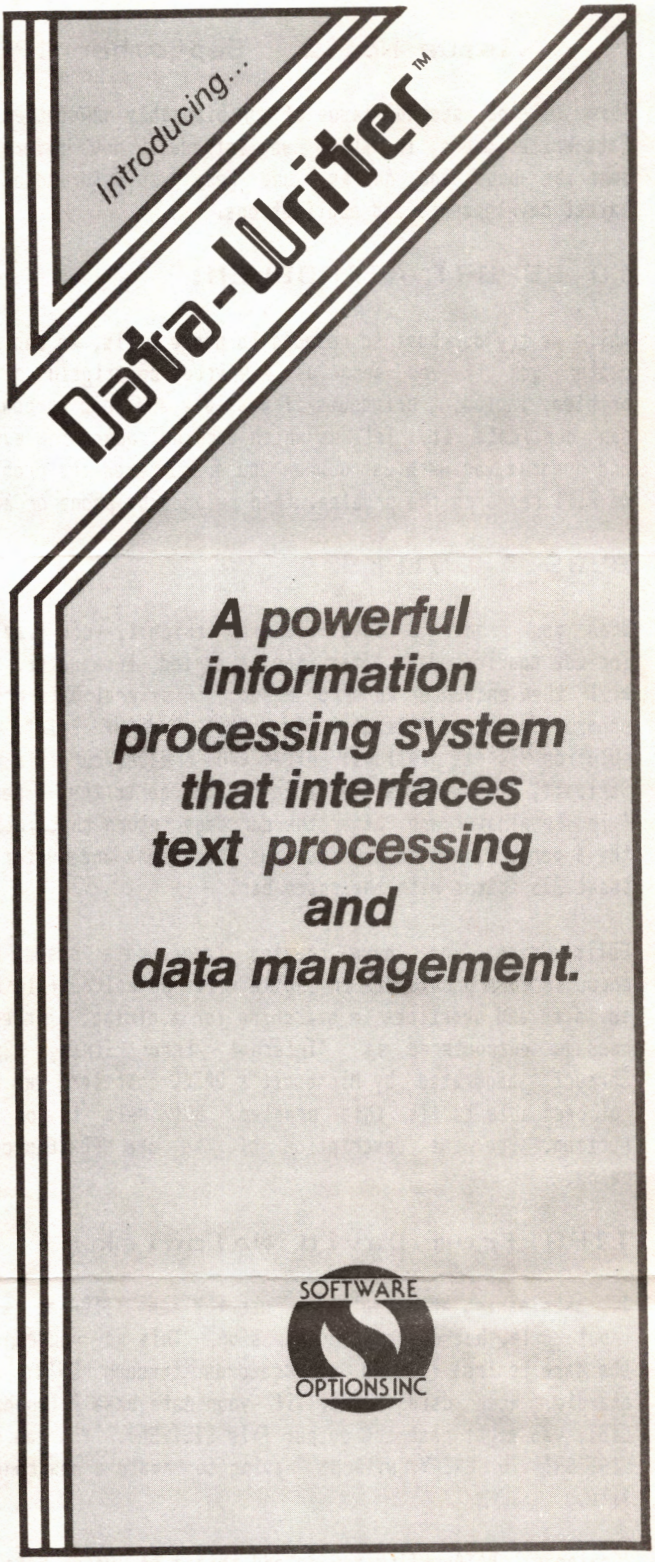
Specify Model I or Model III. If you have a Model III, also specify the operating system under which you run Data-Writer. Shipments will be made the week of November 1st.

NEWS

Watch for Data-Writer reviews upcoming in Computronics, 80-U.S. and 80-Micro. Data-Writer is a hit!

Watch for our ads for Data-Writer/IBM in Softalk/PC and in PC Magazine featuring "The Bug That Ate Boca Raton" and the story of David Walonick's VIP trip to Boca Raton as IBM's guest.

COTS (Commodity Options Trading System) is available now for the IBM PC and Apple II.



SOFTWARE OPTIONS, INC.
19 Rector Street
New York, N.Y. 10006

news for Data-Writer users

Issue No. 2 September 1982

This is the second issue of our bimonthly newsletter for Data-Writer users, in which we introduce new offerings, describe bugs and quirks, and keep you informed of the latest developments and applications.

TO REPORT A PROBLEM:

While we try our best to respond to phone calls, we can do a better job if you send us a written description of the problem, including printouts if possible and info on how we can duplicate it. Tell us which machine, operating system and program you were using when you encountered the problem. We will research the problem, then respond by phone or mail.

BUGS & QUIRKS

When you create a data file with Scripsit, your file may include spurious data after the 4-period terminator. You will then encounter an error message in several Data-Writer programs such as "Attempt to Read Past End of File". The solution is as follows: After you create your file with Scripsit, load your data file into EDIT, go to the 4-period file terminator and, after the carriage return that follows the 4 periods, delete any spurious characters and enter at least 256 spaces with the space bar.

EDIT: Users who have created large data bases have encountered a problem in trying to write an edited file back to disk and sometimes in searching for a string. The error message encountered is "Internal Error--String Space Corrupt" generated by Microsoft's BASIC Compiler. We have not been able to fix this problem. But help is on the horizon. See the description of the new UPDATE program below.

TIPS from David Walonick

1. When naming an output file, use the same filename as the input file, but change the extension. This way you can use the same control file to add records through ENTRY. For example, when using SORT, if your data base filename is LIST, you might name the output file LIST/SRT. You can call LIST/SRT in ENTRY without having to create a new control file.

(Reminder: DATA-WRITER may do odd things to your data if you use the same filename AND same extension for input and output files (in MANAGE, MATH, SORT and SELECT). When using EDIT, your output file may have the same name as your input file provided the entire data base will fit in memory at one time.)

2. You can SORT a data base larger than half your diskette if you do the following. After loading the SORT program, remove the program disk from Drive 0 and insert a minimum system diskette into Drive 0. Write the output file to Drive 0. This can also be done when using EDIT. (To create a minimum system diskette if you are running DATA-WRITER on the Model III under DOSPLUS, make a backup copy of the DATA-WRITER diskette, and then KILL all the DATA-WRITER programs. Remember that for each /CMD file listed in the directory, there is also a /CHN file which is invisible but must also be killed.)

3. When you want to make a change to your control file, you can use EDIT to edit it, rather than creating a new one (if you are very, very careful). The control file is stored in a coded format. Here's the code:

The first number is the number of fields. This is followed in subsequent lines by the names of your field-labels and control information associated with each field (four numbers). The first of the four numbers is the field-length. The second is coded 1 for mandatory entry or 0 for non-mandatory. The third is the coded validity check (see menu of User's Manual, p.18). The fourth number is coded 1 for repeat previous record entry or 0 for non-repeat. The last part of the control file is the Abbreviation Table. Each abbreviation has three parts: the field number that the abbrevation applies to, the abbreviation and the expansion.

DATA-WRITER SUPPORTS MULTIDOS

Here's the procedure to convert the Model III DOSPLUS version to MultiDOS

1. Create two minimum MultiDOS system diskettes.
2. Copy all the Data-Writer A diskette programs to the first MultiDOS diskette, copying both the /CMD files and the /CHN files, which are invisible. For example,

```
COPY EDIT/CMD:1 :0
COPY EDIT/CHN:1 :0
```

3. Type COPY BRUN/CMD:1 :0.
4. Type COPY DIR/CMD:1 TO MENU/CMD:0.
5. Type AUTO MENU.
6. Repeat the same procedure for the Data-Writer B diskette and the second MultiDOS diskette.

SOFTWARE OPTIONS INC ™

To order

Update

phone (800) 221-1624
or (212) 785-8285 or
mail this order form.

Order date .....

Name .....
Address .....
.....
City ..... State .....
Zip ..... Phone ( ).....

Please send me a copy of UPDATE for

☐ Model 1

☐ Model 3 DOSPLUS

☐ Model 3 LDOS

☐ Model 3 NEWDOS/80

UPDATE program $20.00
Diskette (3.00 or enclose diskette). . . . .
For delivery in NY State, add sales tax . . ..
Shipping (2.00 US and Cda, 5.00 foreign) . . ..
TOTAL . . . .

☐ My check is enclosed.

Charge to my: ☐ Mastercard ☐ Visa

Account # .....
Expiration date: .....
Signature .....

SOFTWARE OPTIONS, INC.
19 Rector Street
New York, N.Y. 10006